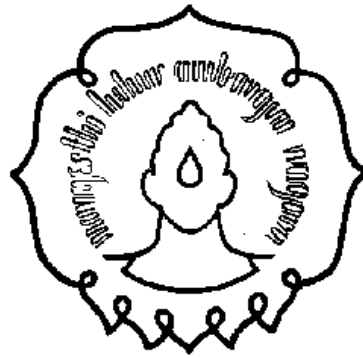


**SENSOR PING PARALLAX SEBAGAI PENGUKUR JARAK  
PADA ROBOT CERDAS PEMADAM API  
BERBASIS MIKROKONTROLER ATMEGA8535**

Diajukan Untuk Memenuhi Sebagian Persyaratan Memperoleh Gelar Ahli Madya  
Program Diploma III Ilmu Komputer



Diajukan Oleh :  
**WAHYU TRI SUTRISNO**  
**M3307063**

**PROGRAM DIPLOMA III ILMU KOMPUTER**  
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**UNIVERSITAS SEBELAS MARET**  
**2010**

**HALAMAN PERSETUJUAN**

**SENSOR PING PARALLAX SEBAGAI PENGUKUR JARAK  
PADA ROBOT CERDAS PEMADAM API  
BERBASIS MIKROKONTROLER ATMEGA8535**

Disusun Oleh

**WAHYU TRI SUTRISNO**

NIM. M3307063

Laporan Tugas Akhir ini disetujui untuk dipertahankan

Di hadapan dewan penguji

pada tanggal 20 Juli 2010

**Dosen Pembimbing**

Muhammad Asri Syafi'ie, S.Si

NIDN. 0603118103

## **HALAMAN PENGESAHAN**

### **SENSOR PING PARALLAX SEBAGAI PENGUKUR JARAK PADA ROBOT CERDAS PEMADAM API BERBASIS MIKROKONTROLER ATMEGA8535**

Disusun Oleh  
WAHYU TRI SUTRISNO  
NIM. M3307063

Dibimbing Oleh  
Pembimbing Utama

Muhammad Asri Syafi'ie, S.Si

Tugas Akhir ini telah diterima dan disahkan oleh Dewan Penguji Tugas Akhir  
Program Diploma III Ilmu Komputer  
Pada tanggal 20 Juli 2010

Dewan Penguji :

1. Penguji 1 : Muhammad Asri Syafi'ie, S.Si ( )  
NIDN. 0603118103
2. Penguji 2 : Rudi Hartono, S.Si ( )  
NIP. 0629088001
3. Penguji 3 : Mohtar Yunianto, S.Si, M.Si ( )  
NIP. 19800630 200501 1 001

Disahkan Oleh

a.n.Dekan FMIPA UNS  
Pembantu Dekan I

Ketua Program Studi  
DIII Ilmu Komputer UNS

Ir. Ari Handono R, M.Sc, Ph.D  
NIP. 19610223 198601 1 001

Drs. YS. Palgunadi, M.Sc  
NIP. 19560407 198303 1 004

## **ABSTRACT**

**WAHYU TRI SUTRISNO. 2010. PING PARALLAX SENSOR AS DISTANCE GAUGE ON FIRE FIGHTING ROBOT BASED ON MICROCONTROLLER ATMEGA8535.** 3<sup>rd</sup> Diploma in Computer Science, Faculty of Mathematics and Natural Science, Sebelas Maret University of Surakarta

Nowadays, robot's development has being used for helping human activity, one of them is firefighting robot. Firefighting robot is a prototype of robot in small size. It was job to extinguishing the fire in a room.

Fire fighting robot requires various sensor to run properly, one of them is using an ultrasonic sensor to navigate the robot. The methods is measuring the distance in the room, get the output sensor as an information and send it to microcontroller. A magnetic compass used to complete this navigation system, by using the magnetic compass we can turn and rotate in a certain angle.

The result is, this robot will search a fire in the room ,using the left-wall method to run it properly and extenguised the fire. The ultrasonic sensors and magnetic compass are used to navigate the robot.

**Keywords:** Ping Parallax sensors, fire fighting, ATMega8535

## ABSTRAK

WAHYU TRI SUTRISNO. 2010. **SENSOR PING PARALLAX SEBAGAI PENGUKUR JARAK PADA ROBOT CERDAS PEMADAM API BERBASIS MIKROKONTROLER ATMEGA8535**. Diploma III Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sebelas Maret Surakarta.

Perkembangan robot pada masa sekarang banyak yang dimanfaatkan untuk membantu tugas manusia, salah satunya robot pemadam api. Robot pemadam api merupakan suatu *prototype* robot dengan ukuran kecil yang diberi tugas memadamkan api lilin dalam suatu ruangan.

Robot pemadam api memerlukan berbagai macam sensor untuk berjalan dengan baik, salah satunya adalah menggunakan sensor ultrasonik yang digunakan untuk navigasi robot. Caranya dengan mengukur jarak pada ruangan, *output* dari sensor sebagai informasi dan mengirimkannya ke mikrokontroler. Magnetic kompas digunakan untuk melengkapi sistem navigasi, dengan menggunakan magnetic kompas kita dapat mengubah dan memutar dalam sudut tertentu.

Hasilnya adalah, robot ini akan mencari api di dalam ruangan, dengan cara menggunakan dinding kiri untuk menjalankannya dengan benar dan mematikan api. Sensor ultrasonik dan magnetic kompas digunakan untuk navigasi robot.

Kata kunci: sensor Ping Parallax, pemadam api, ATmega8535

## MOTTO

Kesalahan kecil bisa mengakibatkan kesalahan yang lebih besar.  
Bersamaan dengan kesalahan itu, persoalannya bisa menjadi besar pula.  
Maka kesalahan kecil pun harus segera dibetulkan.

PENGETAHUAN DAN KETERAMPILAN ADALAH ALAT,  
YANG MENENTUKAN SUKSES ADALAH TABIAT.

*Jadilah dirimu sendiri dan banggalah  
dengan apa yang kamu miliki.*

## **PERSEMBAHAN**

*Karya kecil ini persembahkan untuk:*

*Almamater Universitas Sebelas Maret.*

*Kampus FMIPA Universitas Sebelas Maret.*

*D3 Ilmu Komputer Universitas Sebelas Maret.*

*Keluarga besar ku yang selalu memberi dukungan moril dan materiil.*

*Kekasih yang menyentuh jiwa dan raga ini dengan jemari hati.*

*Teman-teman seperjuangan D3 Teknik Komputer.*

*Rekan satu tim ku, Harnan S.A.*

*Semua pihak yang telah membantu penyusunan tugas akhir.*

## **KATA PENGANTAR**

Segala puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas berkat, rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dan menyusun laporan Tugas Akhir dengan judul **SENSOR PING PARALLAX SEBAGAI PENGUKUR JARAK PADA ROBOT CERDAS PEMADAM API BERBASIS ATMEGA8535**.

Laporan Tugas Akhir ini ditulis untuk memenuhi salah satu persyaratan menempuh Diploma III Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret Surakarta.

Penulis ingin mengucapkan terima kasih kepada :

1. Drs. Y. S. Palgunadi, M.Sc selaku Ketua Program Diploma III Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret Surakarta.
2. Muhammad A. Syafi'ie, S.Si selaku dosen pembimbing yang telah banyak memberikan pengarahan dan saran..
3. Bapak dan Ibu Dosen Program Diploma III Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret Surakarta.
4. Ayah, Ibu, Saudara dan Pacarku, penulis ucapkan terima kasih atas doa dan bantuannya.
5. Teman-teman Diploma III Teknik Komputer UNS angkatan 2007.
6. Saudara Harnan Sholichul A. sebagai rekan satu team pembuatan Tugas Akhir.
7. Berbagai pihak yang tidak dapat penulis sebutkan satu per satu.



Akhirnya, “Tiada Gading Yang Tak Retak “. Penulis menyadari bahwa laporan Tugas Akhir ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran yang bersifat membangun akan penulis terima dengan ikhlas. Semoga laporan ini dapat bermanfaat bagi penulis pada khususnya dan bagi pembaca pada umumnya. Amin

Surakarta, Juni 2010

Penulis

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PERSETUJUAN .....	ii
HALAMAN PENGESAHAN .....	iii
HALAMAN ABSTRACT .....	iv
HALAMAN ABSTRAK .....	v
HALAMAN MOTTO .....	vi
HALAMAN PERSEMBAHAN .....	vii
HALAMAN KATA PENGANTAR .....	viii
HALAMAN DAFTAR ISI .....	x
HALAMAN DAFTAR GAMBAR .....	xii
HALAMAN DAFTAR TABEL .....	xiv
HALAMAN DAFTAR LAMPIRAN .....	xv
<b>BAB I    PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan dan Manfaat Penelitian .....	2
1.4.1 Tujuan Penelitian .....	2
1.4.2 Manfaat Penelitian .....	3
1.5 Metode Penelitian .....	3
1.6 Sistematika Laporan .....	3
<b>BAB II    LANDASAN TEORI .....</b>	<b>5</b>
2.1 Robotik .....	5
2.2 Mikrokontroler ATmega8535 .....	7
2.3 Sistem Gerak Roda Tank .....	10
2.4 Sensor Ping Parallax .....	10
2.5 Sensor UVTron .....	11
2.6 Sensor Garis .....	12
2.7 LCD .....	13

2.8 Motor Servo .....	14
2.9 Motor DC .....	15
2.10 Driver Motor DC .....	16
2.11 Software Pemograman dan Software Downloader .....	16
2.11.1 Men-download-kan program ke mikrokontroler	
Dengan CVAVR .....	17
<b>BAB III DESAIN DAN PERANCANGAN .....</b>	<b>20</b>
3.1 Perancangan Sistem Mekanik .....	20
3.2 Sistem Blok .....	21
3.2.1 Rangkaian Keseluruhan Sistem .....	21
3.2.2 Rangkaian Minimum ATmega8535 .....	22
3.2.3 Rangkaian Sensor Ping .....	24
3.2.4 Rangkaian Sensor Kompas .....	27
3.2.5 Rangkaian Sensor Garis .....	29
3.2.6 Rangkaian Motor DC dan Driver L293D .....	29
3.2.4 Rangkaian Power Supply .....	31
3.3 Perancangan Software .....	32
3.3.1 Flowchart Program Robot Pemadam Api .....	32
<b>BAB IV IMPLEMENTASI DAN ANALISA .....</b>	<b>36</b>
4.1 Pengujian Jarak oleh Sensor Ping Parallax Ultrasonic	
Range Finder.....	37
4.2 Pengujian Jarak Pembacaan Sensor Garis .....	38
4.3 Analisa Pengujian Sensor Ping Parallax dan Sensor Garis ...	39
4.3.1 Analisa Pengujian Sensor Ping Parallax .....	39
4.3.2 Analisa Pengujian Sensor Garis .....	47
<b>BAB V PENUTUP .....</b>	<b>49</b>
5.1 Kesimpulan .....	49
5.2 Saran .....	49
<b>DAFTAR PUSTAKA .....</b>	<b>51</b>
<b>LAMPIRAN</b>	

## DAFTAR GAMBAR

Gambar 2.1 Ilustrasi penelitian dalam domain robotik .....	5
Gambar 2.2 Blok diagram fungsional ATmega8535 .....	8
Gambar 2.3 Konfigurasi pin ATmega8535 .....	9
Gambar 2.4 Minimum system ATmega8535 .....	10
Gambar 2.5 (a) driver sprocket, (b) idler sprocket dan (c) track .....	10
Gambar 2.6 Sensor Ultrasonik Ping .....	11
Gambar 2.7 Sensor UVTron beserta driver C3704 .....	12
Gambar 2.8 Contoh rangkaian sensor garis .....	13
Gambar 2.9 Motor Servo Standart .....	14
Gambar 2.10 Motor DC .....	15
Gambar 2.11 Konfigurasi pin IC L293D .....	16
Gambar 2.12 Screenshoot CodeVision AVR .....	17
Gambar 2.13 Screenshoot proses compile .....	18
Gambar 2.14 Screenshoot proses build .....	18
Gambar 2.15 Screenshoot proses flash programming .....	19
Gambar 3.1 Desain body robot .....	20
Gambar 3.2 Diagram blok robot pemadam api .....	21
Gambar 3.3 Rangkaian Skematik robot .....	23
Gambar 3.4 Pulsa Ping))) Parallax Ultrasonic Range Finder.....	25
Gambar 3.5 Prinsip kerja sensor jarak ultrasonik .....	26
Gambar 3.6 Rangkaian skematik sensor ping ultrasonik .....	25
Gambar 3.7 CMPS03 Magnetic Compass .....	28
Gambar 3.8 Rangkaian skematik sensor kompas .....	28
Gambar 3.9 Rangkaian sensor garis .....	29
Gambar 3.10 Tata letak motor DC pada robot .....	30
Gambar 3.11 Rangkaian motor DC dengan driver L293D .....	31
Gambar 3.12 Rangkaian power supply .....	32
Gambar 3.13 Bentuk Lapangan Uji Coba Robot .....	33
Gambar 3.14 Flowchart Program Robot Pemadam Api .....	34

Gambar 4.1	Rangkaian skematik sensor ultrasonik Ping.....	36
Gambar 4.2	Rangkaian sensor garis .....	37
Gambar 4.3	Dimensi Sensor Ultrasonik .....	40
Gambar 4.4	Pengujian Ping dengan Halangan Tabung .....	40
Gambar 4.5	Pengujian Ping dengan Halangan Papan .....	40
Gambar 4.6	Flowchart mencari room .....	42
Gambar 4.7	Pengujian Ping Ultrasonic .....	47
Gambar 4.7	Prinsip Kerja Sensor Garis.....	48

## **DAFTAR TABEL**

Tabel 2.1 Konfigurasi pin LCD .....	13
Tabel 4.1 Hasil Pengukuran Jarak Rata-rata dan Persen Kesalahan .....	38
Tabel 4.2 Hasil Pengukuran Jarak Sensor Garis .....	39

## **DAFTAR LAMPIRAN**

1. Listing Program
2. Gambar Robot

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan ilmu pengetahuan dan teknologi yang semakin maju banyak yang dimanfaatkan untuk memenuhi kebutuhan manusia. Perkembangan teknologi yang pesat ini ditandai dengan banyaknya peralatan yang telah diciptakan dan dioperasikan baik secara manual maupun otomatis, bahkan ada yang menggunakan robot sebagai alat bantu. Dimana dengan menggunakan robot, kita dapat melakukan pekerjaan yang rumit dan memerlukan ketelitian yang tinggi.

Robot adalah sebuah alat mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dulu (kecerdasan buatan). Robot biasanya digunakan untuk tugas yang berat, berbahaya, pekerjaan yang berulang dan kotor. (<http://id.wikipedia.org/wiki/Robot>, 2010).

Dalam perkembangannya robot telah dikembangkan untuk dapat membantu manusia dalam melakukan pekerjaan yang rumit, berbahaya dan memerlukan ketepatan dan ketelitian tinggi. Pekerjaan tersebut misalnya memadamkan api di bangunan yang strukturnya tidak stabil sehingga sewaktu-waktu dapat runtuh, dengan menggunakan robot pekerjaan tersebut dapat dilakukan tanpa mengancam nyawa petugas pemadam kebakaran.

Robot pemadam api memerlukan berbagai sensor untuk menjalankan fungsinya dengan baik, salah satunya adalah sensor pengukur jarak robot terhadap dinding pada ruangan. Sensor pengatur jarak ini menggunakan sensor *ping ultrasonic range finder* buatan Parallax. Sensor ini akan mengukur jarak robot terhadap dinding, kemudian memberikan informasi kepada mikrokontroler tentang keberadaan dinding beserta jarak terhadap robot pada suatu ruangan sehingga robot dapat melewati atau menghindari halangan yang ada di depan, serta bagian samping robot.

Berdasarkan masalah tersebut penulis mengambil sebuah judul “SENSOR PING PARALLAX SEBAGAI PENGUKUR JARAK PADA



ROBOT CERDAS PEMADAM API BERBASIS MIKROKONTROLER ATMEGA8535”, dengan adanya aplikasi sensor Ping Ultrasonic ini diharapkan robot dapat mencari sumber api dalam suatu ruangan-ruangan berbentuk persegi tanpa menabrak suatu halangan atau dinding pembatas, sehingga setelah sumber api ditemukan maka robot dapat memadamkannya.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah dijelaskan diatas, maka dapat diambil rumusan masalah tentang bagaimana menggunakan sensor PING PARALLAX sebagai pengukur jarak pada robot cerdas pemadam api berbasis mikrokontroler ATmega8535 agar robot tersebut dapat berjalan tanpa menabrak dinding pembatas.

## **1.3 Batasan Masalah**

Sesuai dengan rumusan masalah yang telah dikemukakan, maka batasan masalah dalam tugas akhir ini adalah penggunaan sensor Ping Parallax sebagai pengukur jarak pada robot cerdas pemadam api berbasis mikrokontroler ATmega8535. Penggunaan sensor mencakup pemahaman prinsip kerja sensor Ping Parallax dan pemogramannya menggunakan bahasa C. Robot ini akan disimulasikan berjalan mencari sumber api dalam suatu tempat berbentuk persegi yang memiliki 1 ruang, dalam hal ini robot akan berjalan mengikuti dinding sebelah kiri dengan jarak  $\pm 7$  cm. Setelah sumber api diketemukan maka robot akan memadamkan sumber api tersebut. Bahasan dari tugas akhir ini mencakup proses perancangan, pembuatan dan pengujian.

## **1.4 Tujuan dan Manfaat Penelitian**

### **1.4.1 Tujuan Penelitian**

Tujuan dari penelitian ini adalah memahami prinsip kerja dari sensor Ping Parallax sebagai pengukur jarak pada robot cerdas pemadam api berbasis ATmega8535 agar robot dapat mencari sumber api dalam suatu ruangan tanpa

menabrak suatu halangan atau dinding pembatas dan kemudian memadamkannya.

#### **1.4.2 Manfaat Penelitian**

Manfaat dari penelitian ini adalah sebagai sebuah aplikasi sensor Ping Parallax pada robot pemadam api untuk simulasi skala kecil yang dapat dikembangkan menjadi suatu aplikasi yang dapat membantu petugas pemadam kebakaran di dunia nyata.

### **1.5 Metode Penelitian**

Dalam pembuatan dan penyusunan tugas akhir ini, penulis menggunakan metode sebagai berikut:

#### **a. Metode Literatur**

Metode ini merupakan metode pengumpulan data dan referensi baik dari media cetak maupun media elektronik yang menunjang dalam penyusunan dan pembuatan tugas akhir ini.

#### **b. Metode Observasi**

Metode ini merupakan metode pengumpulan data dengan cara pengamatan terhadap alat telah dibuat yang memiliki kesamaan dengan alat yang akan dibuat dalam tugas akhir ini.

### **1.6 Sistematika Laporan**

Sistematika penulisan laporan tugas akhir sebagai berikut:

#### **1. BAB I PENDAHULUAN**

Bab ini memuat tentang latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, metode penelitian dan sistematika laporan.

#### **2. BAB II LANDASAN TEORI**

Bab ini memuat tentang referensi penunjang yang menjelaskan tentang fungsi dari perangkat-perangkat yang digunakan dalam pembuatan tugas akhir ini.

### **3. BAB III ANALISA DAN PERANCANGAN**

Bab ini memuat tentang penjelasan mengenai perancangan dari perangkat yang akan dibuat.

### **4. BAB IV HASIL DAN PEMBAHASAN**

Bab ini memuat tentang hasil pengujian dari perangkat yang dibuat beserta pembahasannya.

### **5. BAB V PENUTUP**

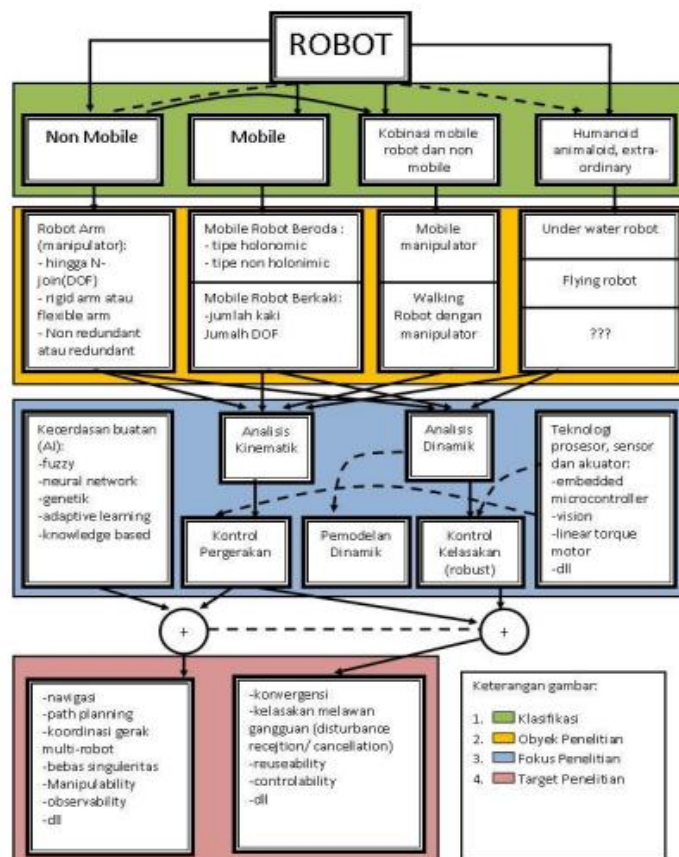
Bab ini memuat tentang kesimpulan dan saran dari pembuatan tugas akhir ini .

## BAB II

### LANDASAN TEORI

#### 2.1 Robotik

Perkembangan suatu ilmu tak lepas dari peran para peneliti kalau tak dapat dikatakan bahwa justru penelitalah yang menyebabkan suatu ilmu itu berkembang. Robotik memiliki unsur yang sedikit berbeda dengan ilmu-ilmu dasar atau terapan yang lain dalam berkembang. Ilmu dasar biasanya berkembang dari suatu hipotesis yang kemudian diteliti secara metodis. Sedangkan ilmu robotik lebih sering berkembang melalui pendekatan praktis pada awalnya. Kemudian melalui suatu pendekatan dari hasil pengamatan perilaku makhluk hidup atau benda bergerak lainnya dikembangkan penelitian secara teoritis



(Sumber: Pitowarno, 2006).

**Gambar 2.1** Ilustrasi penelitian dalam domain robotik

Berdasarkan ilustrasi penelitian dalam domain robotik pada Gambar 2.1, secara garis besar penelitian dibidang robotik dapat dilakukan dengan memilih tema berdasarkan alur dalam 4 tahapan, yaitu klasifikasi, obyek penelitian, fokus penelitian dan target penelitian. Dari blok klasifikasi, struktur robot dapat diketahui berada dalam kelompok mana.

Pada dasarnya dilihat dari struktur dan fungsi fisiknya robot terdiri dari dua bagian, yaitu *non-mobile* robot dan *mobile* robot. Kombinasi keduanya dapat menghasilkan kelompok kombinasi konvensional (*mobile* dan *non-mobile*) dan kelompok non-konvensional. Kelompok konvensional meliputi robot yang memiliki nama-nama yang sudah umum, seperti *mobile manipulator*, *climbing robot*, *walking robot* dan nama-nama yang lain yang sudah populer. Sedangkan kelompok non-konvensional dapat berupa robot *humanoid*, *animaloid*, *extra-ordinary* atau segala bentuk inovasi penyerupaan yang bisa dilakukan.

Kelompok non-mobile yang sering disebut sebagai “keluarga robot” adalah robot *arm* atau *manipulator* saja. Sementara yang lebih mudah dikenali sebagai mesin cerdas (*intelligent machine*) yang tidak selalu tampak memiliki bagian tangan, kaki atau roda untuk bergerak lebih lazim disebut dengan nama khusus sesuai fungsinya, misal *drilling machine*, CNC (*Computer Numerical Control*), EDM (*Electric Discharge Machine*) dan berbagai peralatan otomatis yang biasa dijumpai di pabrik-pabrik modern.

*Mobile* robot adalah tipe robot yang paling populer dalam penelitian robotik. Sebutan ini biasa digunakan sebagai kata kunci utama untuk mencari referensi yang berkaitan dengan robotik di internet. Dari segi manfaat, penelitian tentang berbagai tipe *mobile* robot diharapkan dapat membantu manusia dalam melakukan otomatisasi dalam transportasi, *platform* bergerak untuk robot *industry*, eksplorasi tanpa awak dan masih banyak lagi. (Pitowarno, 2006)

## 2.2 Mikrokontroler ATmega8535

Mikrokontroler AVR (*Alf and Vegard's Risc processor*) memiliki arsitektur 8 bit, dimana semua instruksi dikemas dalam kode 16 bit (*16-bits word*) dan sebagian besar instruksi dieksekusi dalam 1 (satu) siklus *clock*. Mikrokontroler AVR berteknologi RISC (*Reduced Instruction Set Computing*). Secara umum, AVR dikelompokkan menjadi 4 kelas, yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga ATmega dan keluarga AT86RFxx. (Wardhana, 2006)

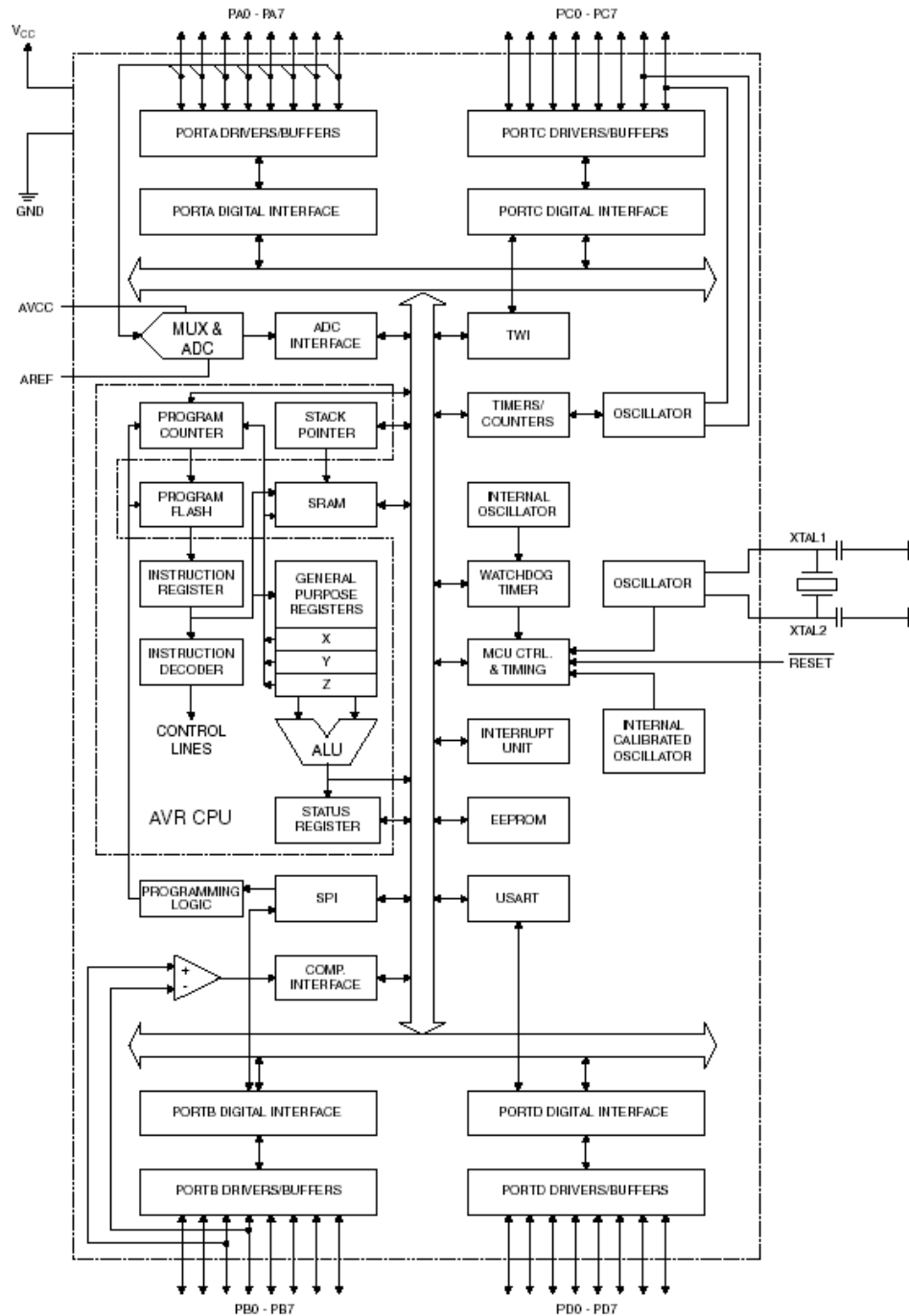
Mikrokontroler AVR ATmega8535 memiliki arsitektur sebagai berikut:

1. Saluran I/O sebanyak 32 buah, yaitu Port A, Port B, Port C dan Port D.
2. ADC 10 bit sebanyak 8 saluran.
3. Tiga buah *Timer/Counter* dengan kemampuan perbandingan.
4. CPU yang terdiri atas 32 register.
5. *Watchdog Timer* dengan osilator internal
6. SRAM sebesar 512 byte
7. Memori *flash* sebesar 8 KB dengan kemampuan *Read While Write*.
8. Unit interupsi internal dan eksternal.
9. Port antarmuka SPI
10. EEPROM sebesar 512 byte yang dapat deprogram saat operasi.
11. Antarmuka komparator analog.
12. Port USART untuk komunikasi serial.

Mikrokontroler AVR ATmega8535 memiliki fitur sebagai berikut:

1. System mikroprosesor 8 bit berbasis RISC dengan kecepatan maksimal 16 MHz.
2. Kapabilitas memori *flash* 8 KB, SRAM sebesar 512 byte dan EEPROM sebesar 512 byte.
3. ADC internal dengan fidelitas 10 bit sebanyak 8 saluran.
4. Portal komunikasi serial (USART) dengan kecepatan maksimal 2,5 Mbps.
5. Enam pilihan mode *sleep* untuk menghemat penggunaan daya listrik.

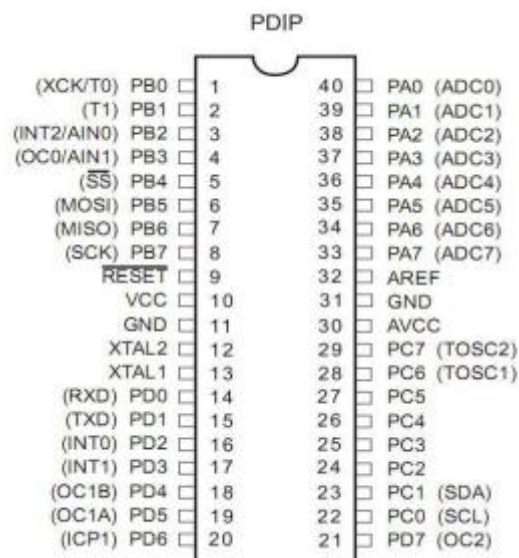
Blok diagram fungsional mikrokontroler ATmega8535 ditunjukan pada Gambar 2.2



**Gambar 2.2** Blok diagram fungsional ATmega8535

Konfigurasi pin dari mikrokontroler ATmega8535 sebanyak 40 pin dapat dilihat pada Gambar 2.3. Dari gambar tersebut dapat dijelaskan secara fungsional konfigurasi pin ATmega8535 sebagai berikut:

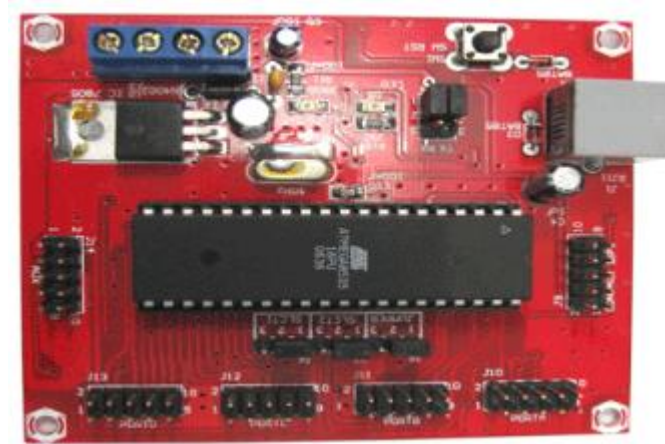
1. VCC merupakan pin yang berfungsi sebagai pin masukan catu daya.
2. GND merupakan pin *ground*.
3. Port A (PA0..PA7) merupakan pin I/O dua arah dan pin masukan ADC.
4. Port B (PB0..PB7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu *Timer/Counter*, komparator analog dan SPI.
5. Port C (PC0..PC7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu TWI, komparator analog dan *Timer Oscillator*.
6. Port D (PD0..PD7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu komparator analog, interupsi eksternal dan komunikasi serial.
7. RESET merupakan pin yang digunakan untuk me-reset mikrokontroler.
8. XTAL1 dan XTAL2 merupakan pin masukan *clock* eksternal
9. AVCC merupakan pin masukan tegangan untuk ADC.
10. AREF merupakan pin masukan tegangan refensi ADC.



**Gambar 2.3** Konfigurasi pin ATmega8535



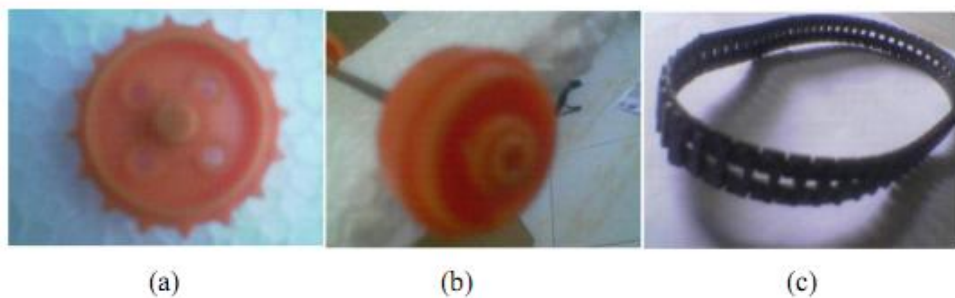
Skema *minimum system* ATmega8535 seperti ditunjukkan pada Gambar 2.4



**Gambar 2.4** *Minimum system ATmega8535*

### 2.3 Sistem Gerak Roda Tank

Sistem gerak roda tank merupakan kombinasi dari tiga bagian, yaitu *drive sprocket*, *idler sprocket* dan *track*. *Drive sprocket* adalah *sprocket* yang berhubungan langsung dengan motor, *idler sprocket* adalah *sprocket* yang membantu pergerakan *track*. *Track* merupakan bagian yang menghubungkan *idler sprocket* dan *drive sprocket* serta mempunyai kontak langsung terhadap permukaan lantai yang dilaluinya. Berikut *drive sprocket*, *idler sprocket* dan *track* ditunjukkan pada Gambar 2.5 (Averroes, 2009)



**Gambar 2.5** *Driver sprocket, idler sprocket dan track*

### 2.4 Sensor Ping Parallax

Sensor Ping merupakan sensor jarak ultrasonik buatan Parallax. Sensor ini bekerja pada frekuensi 40 KHz dan hanya memiliki 3 jalur pin, yaitu

jalur sinyal (SIG), jalur VCC 5V dan jalur *ground*. Sensor Ping mendeteksi jarak objek dengan cara memancarkan gelombang ultrasonik (40 KHz) selama 200 $\mu$ s, kemudian mendeteksi pantulannya. (Budiharto, 2008)

Spesifikasi sensor Ping adalah sebagai berikut:

- Kisaran pengukuran 3cm – 3m
- Input *trigger-positive TTL pulse*, minimal 2 $\mu$ s, tipikal 5 $\mu$ s
- Echo hold off* 750 $\mu$ s dari *fall of trigger pulse*
- Waktu tunda untuk pengukuran selanjutnya 200 $\mu$ s
- Indikator LED untuk aktifitas sensor.



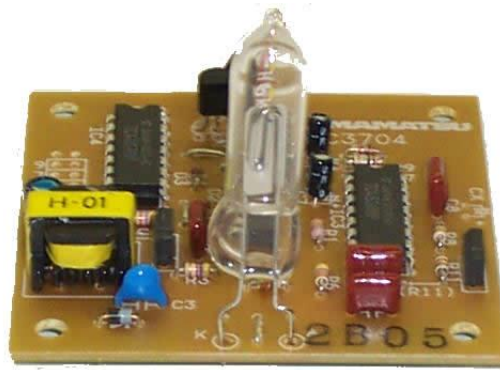
**Gambar 2.6** Sensor Ultrasonik Ping

## 2.5 Sensor UVTron Flame Detector

Hamamatsu UVTron *Flame Detector* dan rangkaian *driver*-nya dapat mendeteksi api dari lilin atau punting rokok dalam jarak 5 meter. Sensor ini digunakan sebagai alat untuk mendeteksi sumber api yang beroperasi pada panjang spectral 185 nm hingga 160 nm.

Tabung UVTron bekerja ketika pada katoda diberikan sinyal ultraviolet, fotoelektron akan dipancarkan dari katoda oleh efek fotoelektrik dan dipercepat ke anoda menggunakan medan listrik. Ketika tegangan diberikan, medan magnet meningkat, medan listrik semakin besar dan energi kinetik elektron menjadi besar untuk mengionisasi molekul gas di tabung untuk bertubrukan. Elektron yang dibangkitkan oleh ionisasi dipercepat yang membuat elektron mengionisasi molekul lainnya sebelum mencapai anoda. Ion positif yang dipercepat ke katoda dan bertubrukan

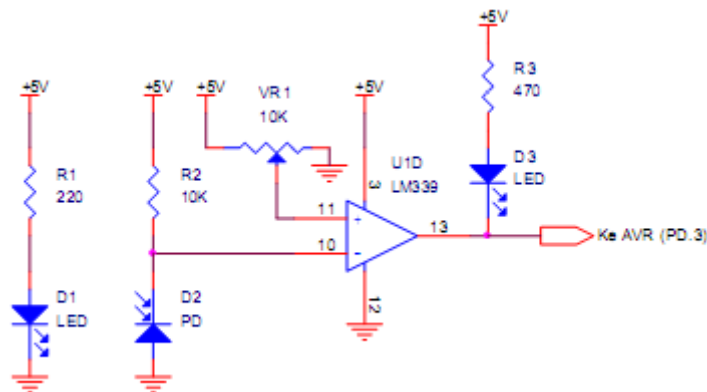
menyebabkan elektron sekunder lainnya, kejadian ini menyebabkan arus yang cukup besar diantara elektroda dan terjadi pembuangan muatan. Berikut bentuk sensor UVTron beserta *driver*-nya ditunjukkan pada Gambar 2.7. (Budiharto, 2008)



**Gambar 2.7** *Sensor UVTron beserta driver C3704*

## 2.6 Sensor Garis (*White Floor Detector*)

Sensor garis ini digunakan untuk membedakan warna hitam atau warna putih. Prinsip dari sensor ini adalah ditembakkannya sinar *infrared* (IR) ke area yang akan dideteksi, jika warna area putih, sebagian sinar IR tersebut akan dipantulkan, sedangkan jika area tersebut berwarna hitam sinar IR akan diserap. Pilihan untuk variasi sensor ini cukup banyak, misal membuat sendiri pasangan antara LED dengan fotodioda dimana LED sebagai pemancar dan fotodioda sebagai penerima atau kombinasi antara LED dengan fototransistor, prinsip kerjanya sama dengan kombinasi LED-fotodioda. Sensor ini harus dipasang dengan jarak kira-kira 1 cm dari atas lantai agar dapat bekerja dengan baik. Salah satu contoh rangkaian sensor garis ditunjukkan pada Gambar 2.8. (Budiharto, 2008)



**Gambar 2.8** Contoh rangkaian sensor garis

## 2.7 LCD

LCD (*Liquid Crystal Display*) adalah sebuah alat yang berfungsi untuk menampilkan huruf, angka atau symbol-simbol tertentu. Tipe LCD yang sering digunakan adalah LCD 16 x 2 (16 kolom 2 baris) dan LCD 20 x 2 (20 kolom 2 baris). (Purwantoni, 2010)

Dalam robotik LCD digunakan sebagai alat penampil pembacaan sensor-sensor pada robot. Dalam pengoperasian LCD ada tiga buah *line control*, yaitu *line EN*, *line RS* dan *line RW*, jika LCD dioperasikan sebagai mode 4 bit maka diperlukan 7 buah *line* (3 *line control* dan 4 *data bus*). Sedangkan jika dioperasikan sebagai mode 8 bit diperlukan 11 *line* (3 *line control* dan 8 *data bus*). Tabel 2.1 menunjukkan konfigurasi pin dari LCD.

**Tabel 2.1** Konfigurasi pin LCD

No. Pin	Nama	Fungsi
1	Vss	Ground
2	Vdd	Positif Supply
3	Vee	Contrast
4	RS	Register Select
5	RW	Read / Write
6	E	Enable
7-14	D0-D7	Data Bus

## 2.8 Motor Servo

Motor *servo* biasanya digunakan untuk robot berkaki, berlengan atau sebagai actuator pada *mobile* robot. Motor *servo* terdiri dari sebuah motor DC, beberapa *gear*, sebuah potensiometer, sebuah *output shaft* dan sebuah rangkaian control elektronik. Motor *servo* dikemas dalam bentuk segi empat dengan sebuah *output shaft* motor dan konektor dengan 3 kabel yaitu *ground*, *power* dan *control*.

Jenis motor *servo* berdasarkan sudut operasi motor *servo* dibagi menjadi 2 yaitu:

a. Motor Servo Standart

Motor *servo standart* merupakan motor *servo* yang mampu bergerak CW dan CCW dengan sudut operasi tertentu, misal  $60^\circ$ ,  $90^\circ$  atau  $180^\circ$ . sudut maksimal yang diperbolehkan untuk motor *servo* standart adalah  $180^\circ$ . Motor *servo* ini sering dipakai pada sistem robotika yang menggunakan lengan atau kaki.

b. Motor Servo Continuous

Motor *servo continuous* adalah motor *servo* yang mampu bergerak CW dan CCW tanpa batasan sudut operasi (berputar secara kontinyu). Motor *servo* ini sering digunakan sebagai aktuator pada *mobile* robot.

Motor *servo* beroperasi pada tegangan *supply* 4,8 volt hingga 7,2 volt. Bentuk fisik motor *servo* ditunjukkan pada Gambar 2.9. (Averroes, 2009)



**Gambar 2.9** Motor Servo Standart

## 2.9 Motor DC

Motor DC adalah suatu motor penggerak yang dikendalikan dengan arus searah (DC). Bagian motor DC yang paling penting adalah *rotor* dan *stator*, yang termasuk *stator* adalah badan motor, sikat-sikat dan inti kutub magnet. Bagian *rotor* adalah bagian yang berputar dari motor DC, yang termasuk *rotor* ialah lilitan jangkar, jangkar, *komutator*, tali, *isolator*, poros, bantalan dan kipas. Bentuk fisik motor DC ditunjukkan pada Gambar 2.10 (Heryanto dan Adi, 2008)



**Gambar 2.10** *Motor DC*

Menurut Heryanto dan Adi (2008), motor DC dapat dikelompokkan menjadi 5 macam:

a. *Externally-excited DC motor*

Motor DC ini kumparan eksitasi dan jangkar tidak terhubung langsung.

b. *Series DC motor*

Motor DC ini kumparan eksitasi dan jangkarnya terhubung secara seri. Motor ini mempunyai torsi awal yang besar.

c. *Shunt DC motor*

Motor DC ini kumparan eksitasi dan jangkarnya dihitug secara paralel (*shunt*). Motor ini digunakan untuk kecepatan konstan dan berubah-ubah.

d. *Compound DC motor*

Motor DC ini disebut juga *cumulatively-compounded DC motor* karena konstruksi motor DC ini memiliki hubungan seri sekaligus *shunt* dan saling menjumlah.

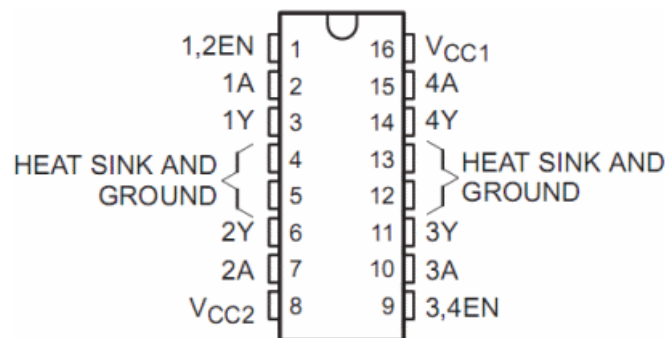
e. *Compounded DC motor*

Motor DC ini disebut juga *differential-compounded DC motor* karena hubungan seri dan *shunt*-nya saling berlawanan.

## 2.10 Driver Motor DC

*Driver* motor digunakan untuk menggerakkan motor DC menggunakan mikrokontroler. Arus yang mampu diterima atau yang dikeluarkan oleh mikrokontroler sangat kecil (dalam satuan miliampere) sehingga agar mikrokontroler dapat menggerakkan motor DC diperlukan suatu rangkaian *driver* motor yang mampu mengalirkan arus sampai dengan beberapa ampere.

Rangkaian driver motor DC dapat berupa rangkaian transistor, *relay*, atau IC (*Integrated Circuit*). Rangkaian *driver* yang umum digunakan adalah dengan IC L293D. IC L293D berisi 4 *channel driver* dengan kemampuan mengalirkan arus sebesar 600mA per *channel*. Tegangan kerja IC L293D dari 6 volt sampai dengan 36 volt dan arus impuls tak berulang maksimum sebesar 1,2 ampere. Konfigurasi pin IC L293D ditunjukkan pada Gambar 2.11. (Wiyono, 2007)



**Gambar 2.11** Konfigurasi pin IC L293D

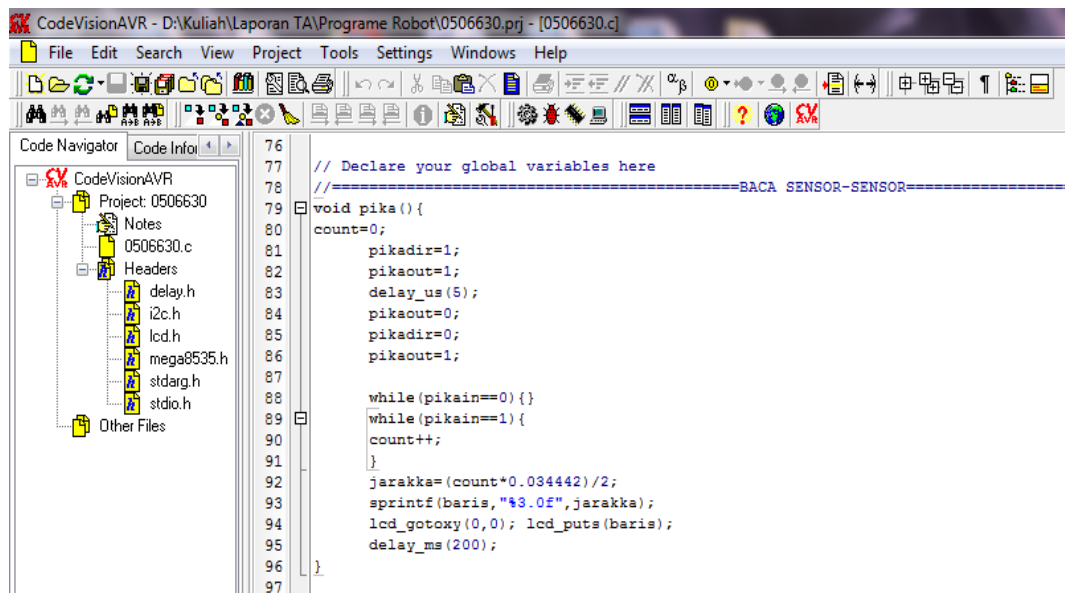
## 2.11 Software Pemograman dan Software Downloader

Bahasa C merupakan salah satu bahasa yang cukup populer dan handal untuk pemograman mikrokontoler. Dalam melakukan pemograman mikrokontroler diperlukan suatu *software* pemograman, salah satunya yang mendukung bahasa C adalah *Code Vision AVR* (CVAVR). CVAVR hanya

dapat digunakan pada mikrokontroler keluarga AVR. CVAVR selain dapat digunakan sebagai *software* pemograman juga dapat digunakan sebagai *software downloader*. *Software downloader* akan men-download-kan file berekstensi “.hex” ke mikrokontroler. (Averroes, 2009)

### 2.11.1 Men-download program ke mikrokontroler dengan CVAVR

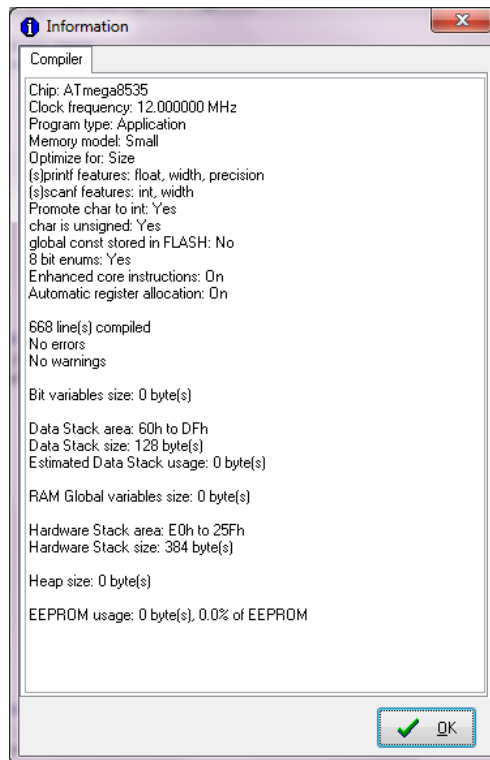
Persiapan pertama sebelum men-download adalah menghubungkan minimum sistem ATmega835 dengan PC melalui USB *port* atau *serial port* tergantung spesifikasi minimum sistemnya. Langkah berikutnya adalah membuat *listing* program yang akan di-download-kan nantinya dengan CVAVR, seperti tampak pada Gambar 2.12.



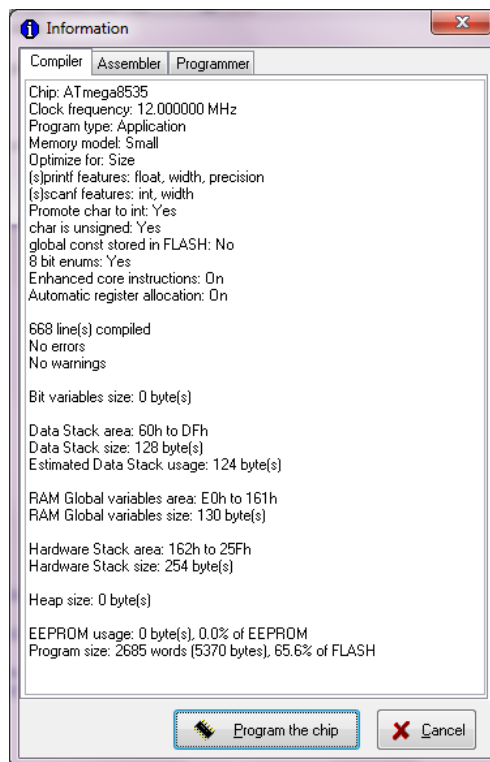
**Gambar 2.12** Screenshot CodeVision AVR

Langkah berikutnya setelah pengetikan *listing* program selesai adalah proses *compile*, yaitu proses pengecekan adanya *error* pada *listing* program yang telah dibuat, jika tidak terdapat *error* seperti pada Gambar 2.13 *listing* program dapat disimpan. Program tersebut akan disimpan dengan ekstensi “.c”, agar dapat di-download ke mikrokontroler maka ekstensi tersebut harus diubah dulu ke ekstensi “.hex”, yaitu dengan cara “*build*” atau dengan kombinasi tombol “Shift+f9”, maka akan tampak seperti pada Gambar 2.14.



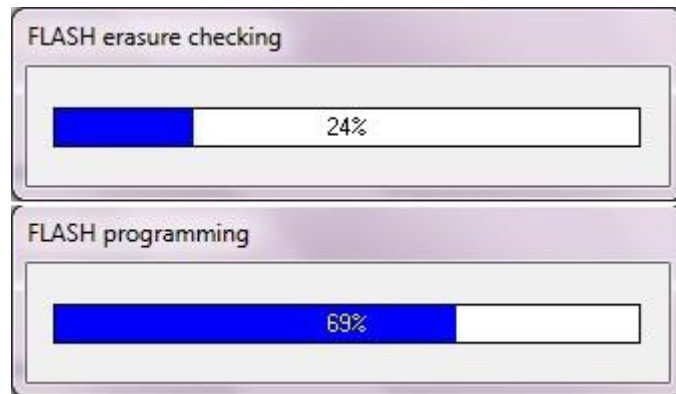


**Gambar 2.13** Screenshoot proses compile



**Gambar 2.14** Screenshoot proses build

Langkah selanjutnya, untuk proses pengisian program ke mikrokontroler ATmega8535 (*flash programming*) yaitu dengan cara menekan tombol “*program the chip*” pada *window build* tadi. Prosesnya seperti pada Gambar 2.15.



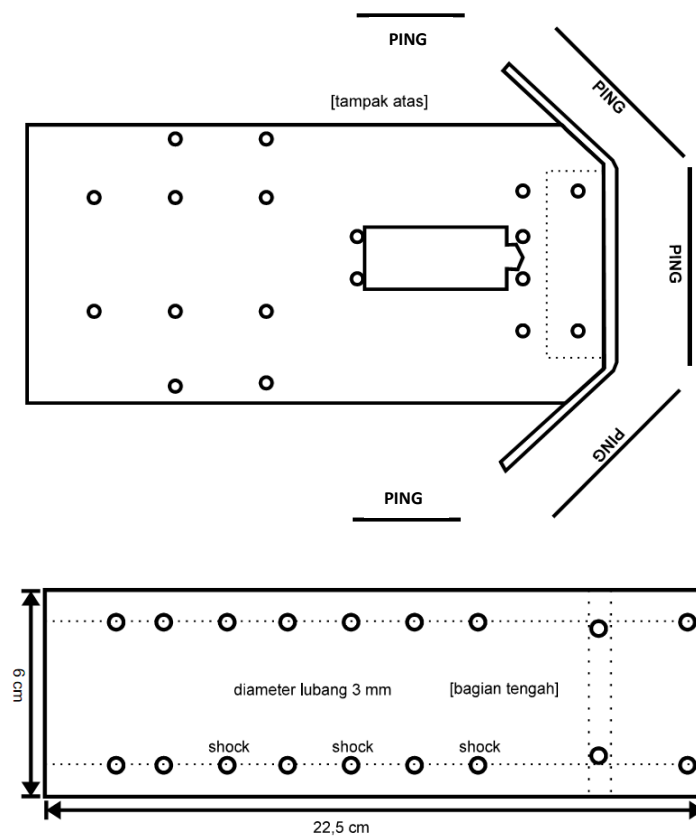
**Gambar 2.15** Screenshot proses flash programming

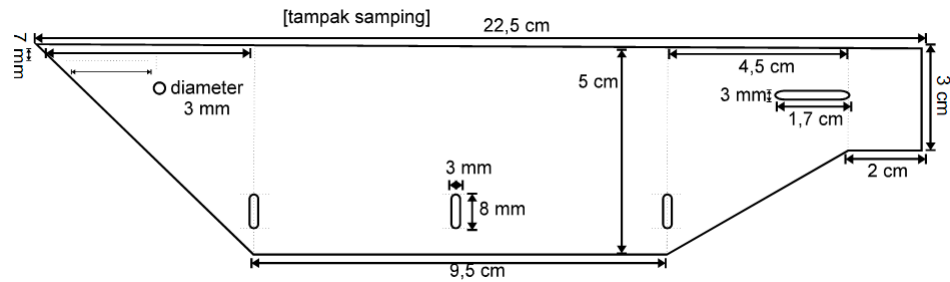
## BAB III

### DESAIN DAN PERANCANGAN

#### 3.1 Perancangan Sistem Mekanik

Robot pemadam api ini memiliki dimensi panjang 24 cm, lebar 15 cm dan tinggi 17 cm. *Body* robot menggunakan bahan dari akrilik setebal 5 mm. *body* robot dibuat menjadi 3 tingkat yaitu tingkat dasar untuk *gearbox* dan roda tank, tingkat pertama untuk peletakan baterai dan tingkat kedua untuk peletakan sensor-sensor dan minimum sistem robot. Desain *body* robot pemadam api ditunjukkan pada Gambar 3.1.

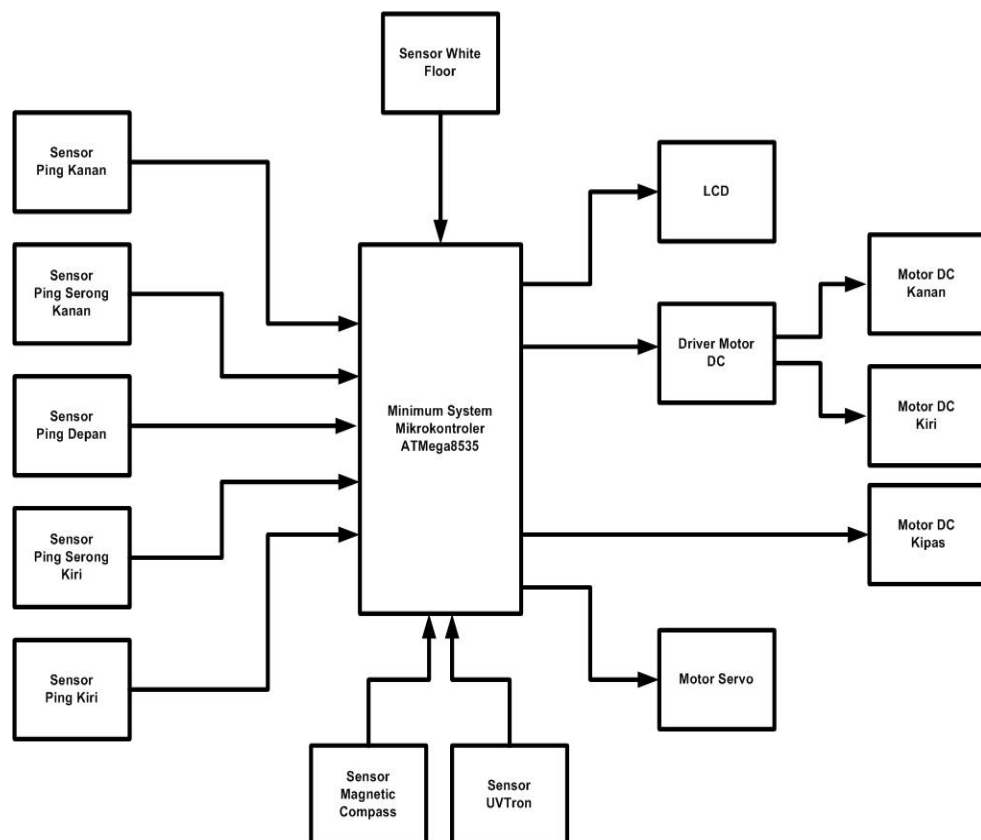




**Gambar 3.1** Desain body robot

### 3.2 Sistem Blok

Robot pemadam ini memiliki perencanaan perangkat keras sebagai berikut: mikrokontroler ATmega8535, 5 buah sensor Ping ultrasonik, 1 buah sensor UVTron *flame detector*, 1 buah sensor *magnetic compass*, 1 buah rangkaian sensor garis, 1 buah motor *servo*, 1 buah LCD 16x2 sebagai penampil pembacaan sensor, 1 buah *gearbox* dengan 2 buah motor dc beserta driver L293D dan 1 buah rangkaian kipas untuk memadamkan api. Blok diagram robot pemadam api ditunjukkan pada Gambar 3.2.



**Gambar 3.2** Diagram blok robot pemadam api

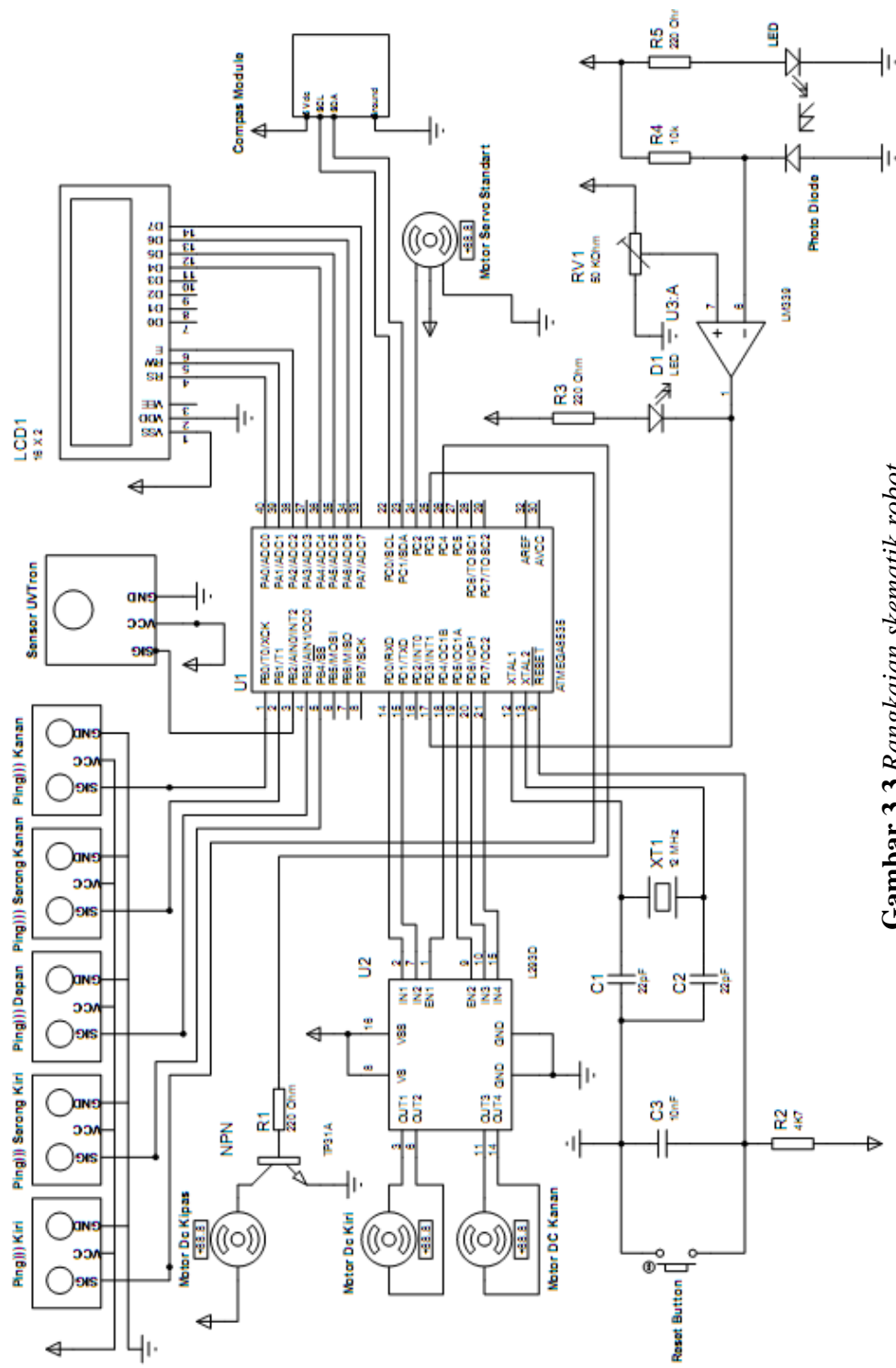
### 3.2.1 Rangkaian Keseluruhan Sistem

Rangkaian keseluruhan sistem dari robot pemadam api dapat dibagi menjadi 4 bagian utama yaitu: bagian minimum sistem mikrokontroler, bagian input, bagian output dan bagian *power supply*. Bagian minimum sistem mikrokontroler merupakan bagian dimana input-input dari sensor akan diproses sesuai program yang digunakan kemudian diteruskan ke bagian output. Bagian input merupakan sensor-sensor yang digunakan untuk pengambilan data terhadap lingkungan yang kemudian akan menjadi dasar dari pergerakan robot. Bagian input terdiri dari 5 buah sensor ping ultrasonik, 1 buah sensor UVTron *flame detector*, 1 buah sensor *magnetic compass*, dan 1 buah rangkaian sensor garis. Bagian output terdiri dari 1 buah LCD 16x2 untuk mengetahui hasil pembacaan sensor, 1 buah motor *servo* untuk menggerakkan kedudukan sensor UVTron 2 buah motor dc beserta *driver*-nya dan 1 buah motor dc dengan kipas untuk memadamkan api. Rangkaian skematik robot pemadam api ditunjukkan pada Gambar 3.3.

### 3.2.2 Rangkaian Minimum ATmega8535

Sistem minimum yang digunakan adalah sistem minimum mikrokontroler ATmega8535 buatan Creative Vision. Sistem minimum ini telah terintegrasi dengan 32 jalur I/O dari mikrokontroler ATmega8535, dapat digunakan untuk mem-*download*-kan program ke mikrokontroler dengan *interfacing* USB serta dapat digunakan dengan *power supply* sebesar 5 volt atau 7-24 volt. Port I/O yang digunakan untuk robot pemadam api sebagai berikut: Port A digunakan untuk *interfacing* LCD 16x2, Port B.0, B.1, B.3, B.4 dan C.3 digunakan untuk *interfacing* dengan sensor Ping Ultrasonik, Port B.2 digunakan untuk *interfacing* dengan sensor UVTron *flame detector*, Port C.4 digunakan untuk *interfacing* dengan rangkaian motor dc kipas, Port C.0 dan C.1 digunakan untuk *interfacing* dengan sensor *magnetic compass*. Port C.2 digunakan untuk *interfacing* dengan motor *servo*, Port D.3 digunakan untuk *interfacing* dengan sensor garis, Port D.0,

D.1, D.4, D.5, D.6, D.7 digunakan untuk *interfacing* dengan IC Driver Motor L293D.



Gambar 3.3 Rangkaian skematik robot

### 3.2.3 Rangkaian Sensor Ping

Ping))) adalah modul pengukur jarak dengan ultrasonic buatan Parallax Inc. yang didesain khusus untuk teknologi robotika. Dengan ukurannya yang cukup kecil (2,1cm x 4,5cm), sensor ini dapat mengukur jarak antara 3 cm sampai 300 cm. Keluaran dari Ping))) berupa pulsa yang lebarnya merepresentasikan jarak. Lebar pulsanya bervariasi dari 115 uS sampai 18,5 mS. Sensor jarak ini digunakan karena hanya membutuhkan satu jalur sinyal selain jalur 5 volt dan *ground*.

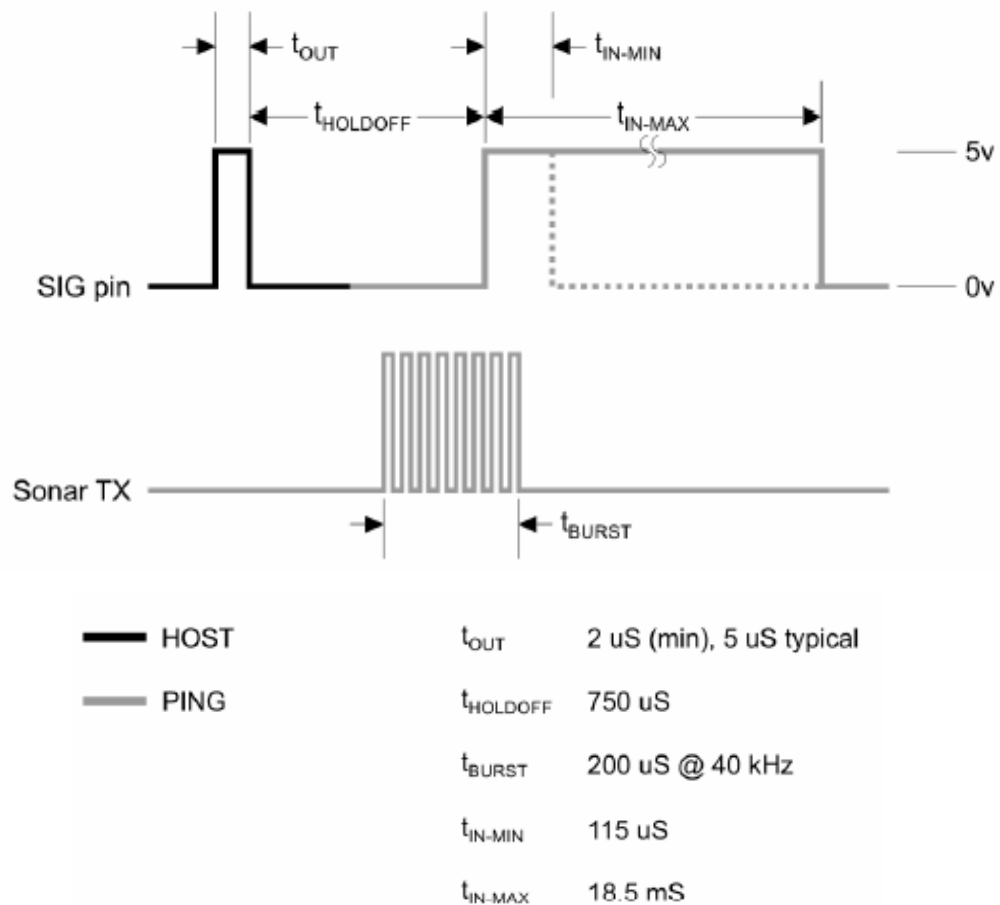
Pada dasarnya, Ping))) terdiri dari sebuah chip pembangkit sinyal 40KHz, sebuah speaker ultrasonik dan sebuah mikropon ultrasonik. Speaker ultrasonik mengubah sinyal 40 KHz menjadi suara sementara mikropon ultrasonik berfungsi untuk mendeteksi pantulan suaranya. Pada modul Ping))) terdapat 3 pin yang digunakan untuk jalur power supply (+5V), ground dan signal. Pin signal dapat langsung dihubungkan dengan mikrokontroler tanpa tambahan komponen apapun.

Ping))) mendeteksi objek dengan cara mengirimkan suara ultrasonik dan kemudian “mendengarkan” pantulan suara tersebut. Ping))) hanya akan mengirimkan suara ultrasonik ketika ada pulsa trigger dari mikrokontroler (Pulsa high selama 5uS). Suara ultrasonik dengan frekuensi sebesar 40KHz akan dipancarkan selama 200uS. Suara ini akan merambat di udara dengan kecepatan 344.424m/detik (atau 1cm setiap 29.034uS), mengenai objek untuk kemudian terpantul kembali ke Ping))). Selama menunggu pantulan, Ping))) akan menghasilkan sebuah pulsa. Pulsa ini akan berhenti (low) ketika suara pantulan terdeteksi oleh Ping))). Oleh karena itulah lebar pulsa tersebut dapat merepresentasikan jarak antara Ping))) dengan objek.

Selanjutnya mikrokontroler cukup mengukur lebar pulsa tersebut dan mengkonversinya dalam bentuk jarak dengan perhitungan sebagai berikut:

$$\text{Jarak} = (\text{Lebar Pulsa} / 29.034 \mu\text{S}) / 2 \text{ (dalam cm), atau}$$

$$\text{Jarak} = (\text{Lebar Pulsa} \times 0.034442) / 2 \text{ (dalam cm), Karena } 1/29.034 = 0.034442$$



**Gambar 3.4** Pulsa Ping))) Parallax Ultrasonic Range Finder

Jarak antara sensor dengan objek yang direfleksikan dapat dihitung dengan menggunakan rumus:

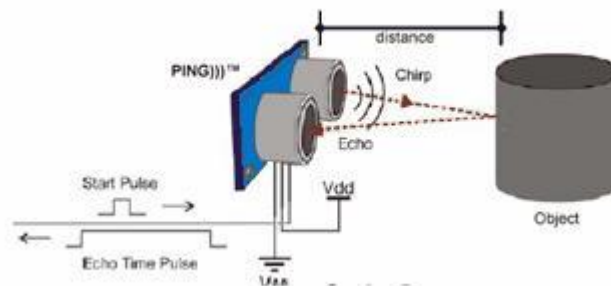
$$L = \frac{1}{2} \cdot \text{TOF} \cdot C \quad (3.1)$$

Dimana :      L      = Jarak Objek  
                  TOF   = Waktu pengukuran yang diperoleh



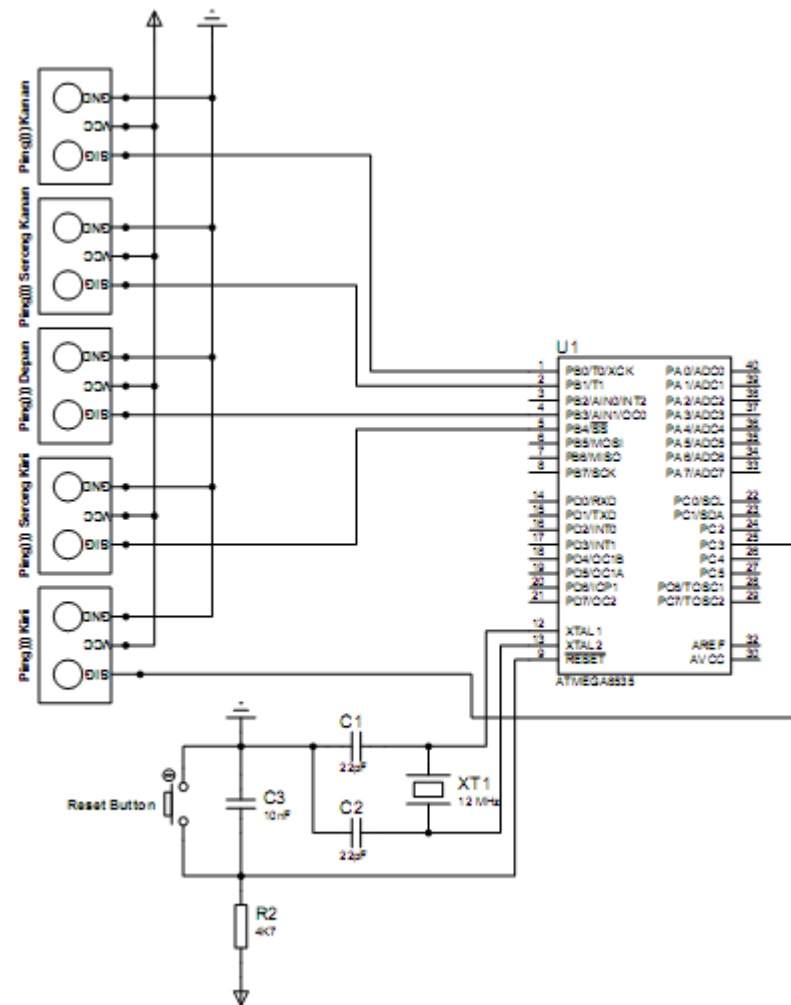
$C$  = Cepat rambat suara (340 m/s )

Sensor jarak ultrasonik ini memiliki kontak jarak 3 cm sampai dengan 3 meter. Ilustrasi dari prinsip kerja sensor jarak ultrasonik Ping ditunjukkan pada Gambar 3.4. (Kurniawan, 2008)



**Gambar 3.5** Prinsip kerja sensor jarak ultrasonik

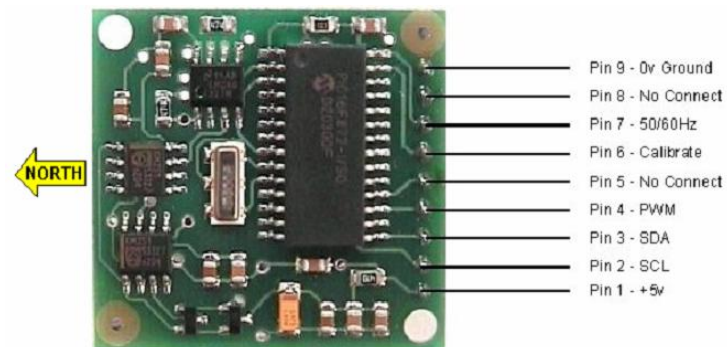
Robot pemadam api ini menggunakan 5 buah sensor jarak ultrasonik Ping yang terletak pada bagian depan, serong kanan, kanan, serong kiri, kiri dari robot. *Interfacing* sensor ultrasonik Ping dengan mikrokontroler ATmega8535 menggunakan Port B.0, B.1, B.3, B.4 dan Port C.3. Sensor yang dipasang dengan posisi serong memiliki derajat keserongan sebesar  $45^0$ , hal ini dimaksudkan agar pembacaan jarak lebih sensitive dan *range* lebih luas. Rangkaian skematik sensor jarak ultrasonik ditunjukkan pada Gambar 3.5.



**Gambar 3.6** Rangkaian skematik sensor ultrasonik Ping

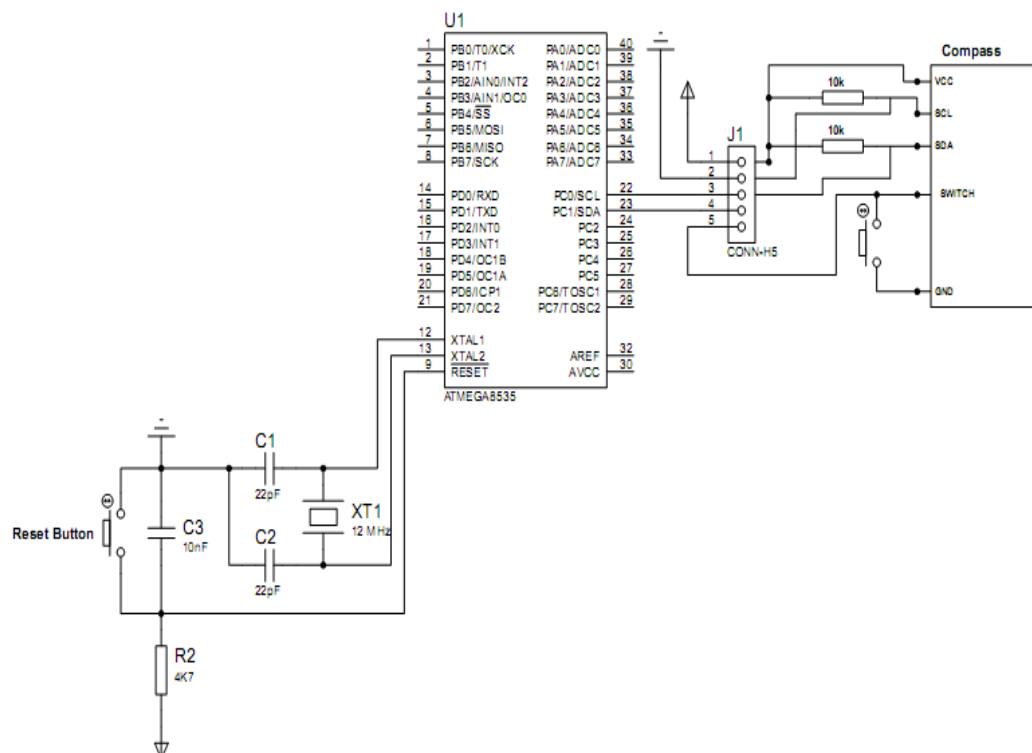
### 3.2.4 Rangkaian Sensor Magnetic Compass

Robot pemadam ini menggunakan sensor CMPS03 Magnetic Compass buatan Devantech Ltd. Sensor ini berukuran 4 x 4 cm dan menggunakan sensor medan magnet Philips KMZ51 yang cukup sensitive untuk mendeteksi medan magnet bumi. Sensor ini memiliki 9 pin yang fungsinya ditunjukkan pada Gambar 3.9.



**Gambar 3.7** *CMPS03 Magnetic Compass*

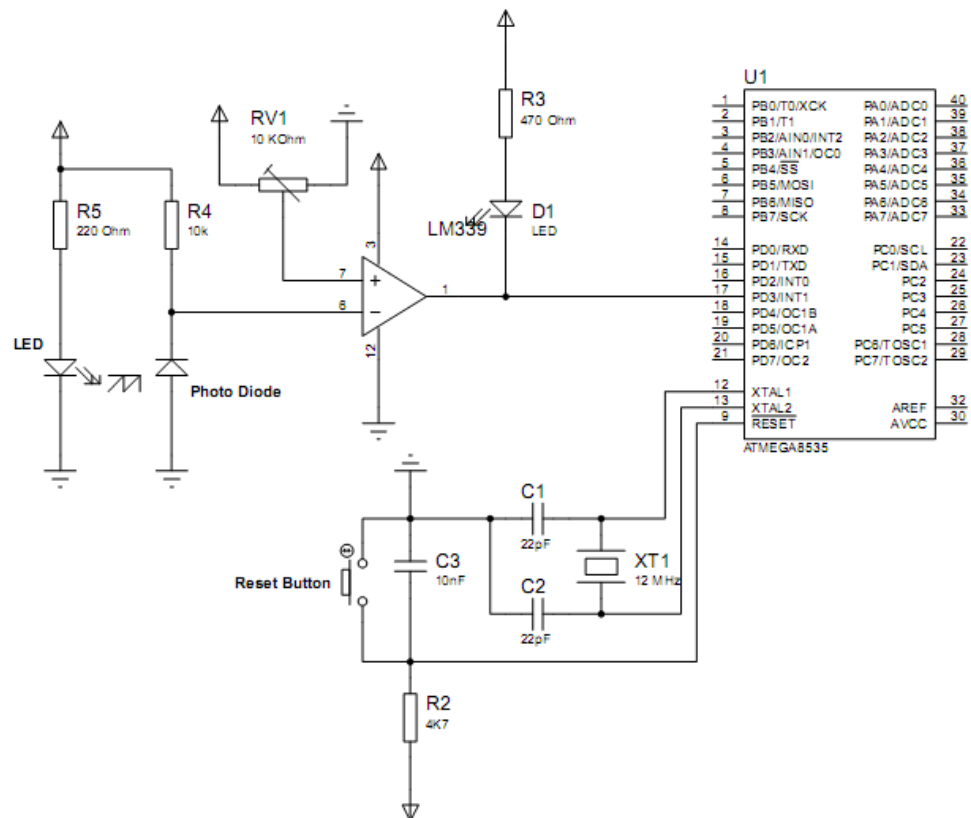
Penggunaan sensor ini dimaksudkan supaya sistem navigasi robot menjadi lebih baik. Karena dapat mengetahui arah mata angin melalui sensor ini. *Interfacing* sensor kompas dengan mikrokontroler menggunakan Port C.0 dan C.1, port tersebut memiliki fungsi khusus sebagai jalur komunikasi I2C, pembacaan sensor ini harus menggunakan jalur tersebut. Rangkaian skematik sensor kompas ditunjukkan pada Gambar 3.10.



**Gambar 3.8** *Rangkaian skematik sensor kompas*

### 3.2.5 Rangkaian Sensor Garis

Robot pemadam ini menggunakan sensor garis untuk mengetahui jarak maksimal antara robot dengan sumber api saat pemadamam api akan dilakukan. Sensor garis yang digunakann pada robot ini merupakan perpaduan antara LED dengan *photodiode*, dimana LED sebagai pemancar dan *photodiode* sebagai penerima. *Interfacing* sensor ini dengan mikrokontroler menggunakan Port D.3. rangkaian skematik sensor garis ini ditunjukkan pada Gambar 3.11.

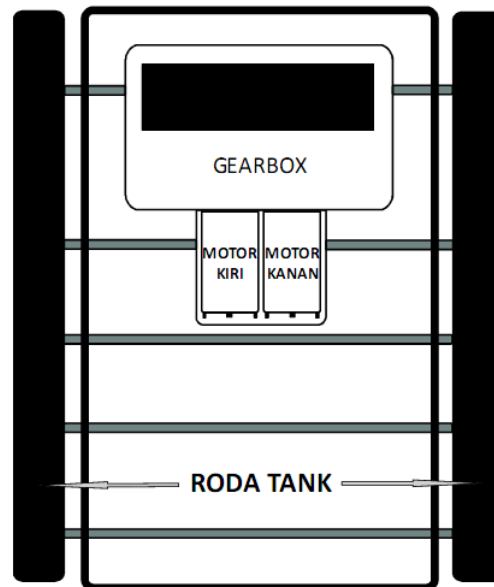


Gambar 3.9 Rangkaian sensor garis

### 3.2.6 Rangkaian Motor DC dan Driver L293D

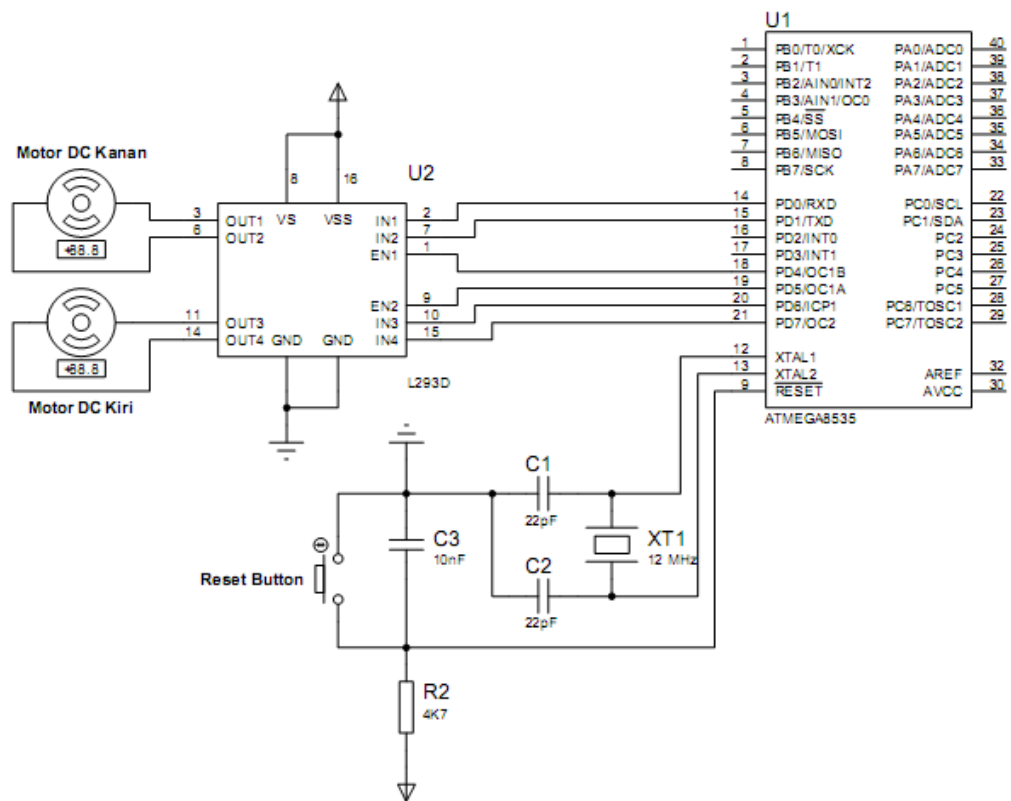
Motor DC yang digunakan berjumlah 2 buah dengan tegangan kerja 3 volt. Motor DC ini diletakkan pada *gearbox* dibawah bodi robot. Motor DC

memiliki 2 input yaitu 3 Vdc dan ground. Tata letak motor DC dan *gearbox* pada robot pemadam ditunjukkan pada Gambar 3.12.



**Gambar 3.10** Tata letak motor DC pada robot

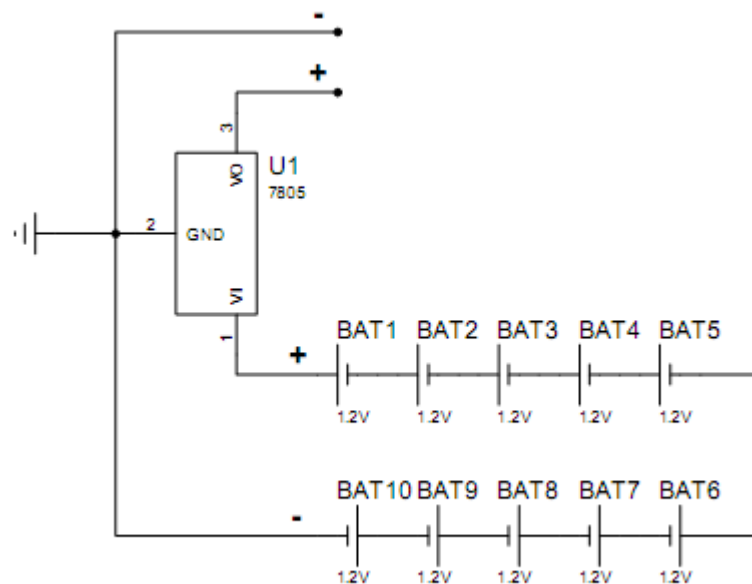
Pergerakan robot tentu saja tidak hanya sebatas maju ke depan, namun dapat mundur, belok kanan dan belok kiri, agar motor dc dapat bergerak seperti itu maka harus menggunakan IC *driver* L293D. IC L293D memiliki 16 pin dengan fungsinya masing-masing, dengan memberikan logika pada masukan IN1, IN2 serta *Enable* maka pergerakan motor DC dapat diatur arahnya maupun kecepatannya. *Interfacing driver* L293D dengan mikrokontroler sebagai berikut: input1 ke Port D.0, input2 ke PortD.1, enable1 ke Port D.4, enable2 ke Port D.5, input3 ke Port D.6 dan input4 ke Port D.7, sedangkan untuk outputnya sebagai berikut: output1 dan output2 ke motor DC 1, output3 dan output4 ke motor DC 2. Rangkaian skematik dari motor DC beserta *driver* L293D ditunjukkan pada Gambar 3.13.



**Gambar 3.11** Rangkaian motor DC dengan driver L293D

### 3.2.7 Rangkaian Power Supply

Bagian *power supply* robot pemadam ini terdiri dari 10 buah baterai Ni-Cad 2700mAH yang disusun secara seri. 1 buah baterai memiliki tegangan 1,2 volt, sehingga total tegangan 12 volt DC. Tegangan 12 Vdc akan diregulasi terlebih dahulu oleh IC 7805 menjadi 5 Vdc. Rangkaian *power supply* ditunjukkan pada Gambar 3.14.



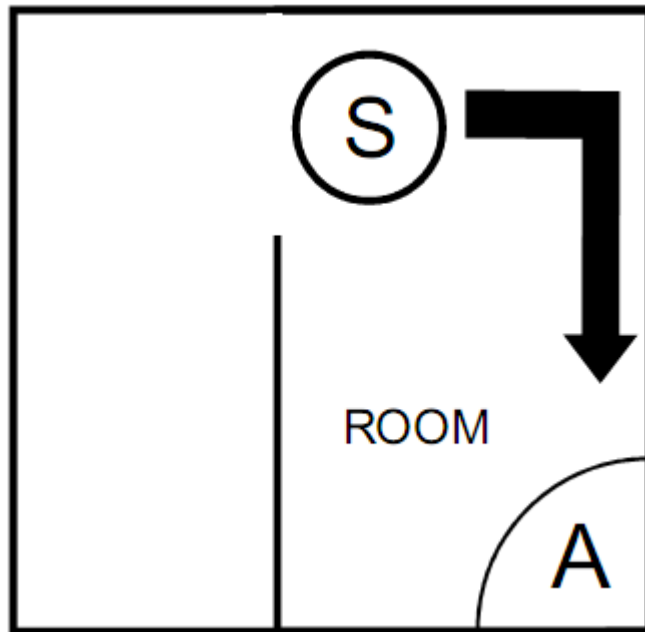
**Gambar 3.12** *Rangkaian power supply*

### 3.3 Perancangan Software

Perancangan software merupakan proses perancangan pembuatan program yang nantinya akan dijalankan oleh mikrokontroler. Program ini nantinya akan menjadi rutin yang akan selalu dijalankan ketika mikrokontroler dinyalakan. Program ini akan disimpan pada EEPROM yang ada didalam mikrokontroler, sehingga hanya perlu sekali *men-download* kan program ke mikrokontroler karena walaupun sumber tegangan dimatikan program masih tersimpan pada EEPROM.

#### 3.3.1 *Flowchart* Program Robot Cerdas Pemadam Api

Program robot pemadam api disesuaikan dengan kondisi lapangan yang akan digunakan, adapun bentuk dan arah gerak robot ditunjukkan pada Gambar 3.13.



**Gambar 3.13** *Bentuk Lapangan Uji Coba Robot*

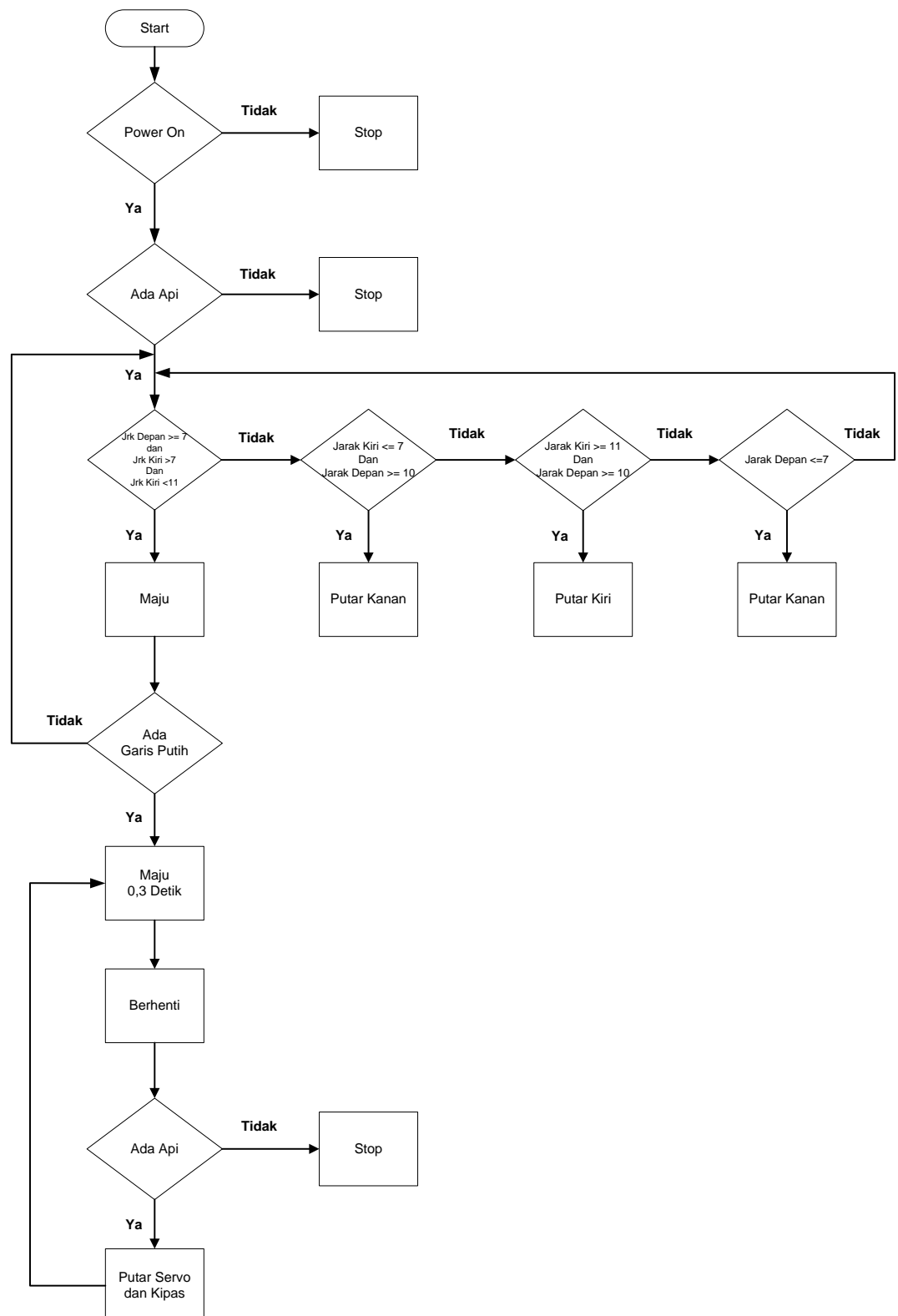
Keterangan gambar :

S = Start Robot

A = Letak Titik Api (Lilin)

Berdasarkan bentuk lapangan dan arah gerak robot dapat dibuat *flowchart* program robot sebagai panduan dalam membuat program robot. *Flowchart* ini akan menunjukkan arah dan tujuan terhadap program yang dibuat untuk robot pemadam api. *Flowchart* program robot adalah sebagai berikut:





**Gambar 3.14** Flowchart Program Robot Pemadam Api

Robot akan menyala jika power dinyalakan, kemudian pertama kali robot melakukan check api, jika tidak ada api robot akan diam, jika terdapat api robot akan melakukan check jarak, jika jarak depan lebih dari 7 dan jarak kiri lebih dari 7 dan kurang dari 11, robot akan bergerak maju, jika tidak terpenuhi check jarak kembali, jika jarak kiri  $\leq 7$  dan jarak depan  $\geq 10$ , robot putar kanan, jika tidak terpenuhi check jarak kembali, jika jarak kiri  $\geq 11$  dan jarak depan  $\geq 10$ , robot putar kiri, jika tidak terpenuhi check jarak kembali, jika jarak depan  $\leq 7$ , robot putar kanan.

Pada saat robot maju dilakukan check garis putih, jika tidak maka robot kembali check jarak, jika terpehuni robot akan maju selama 0,3 detik kemudian berhenti, dan melakukan check api.

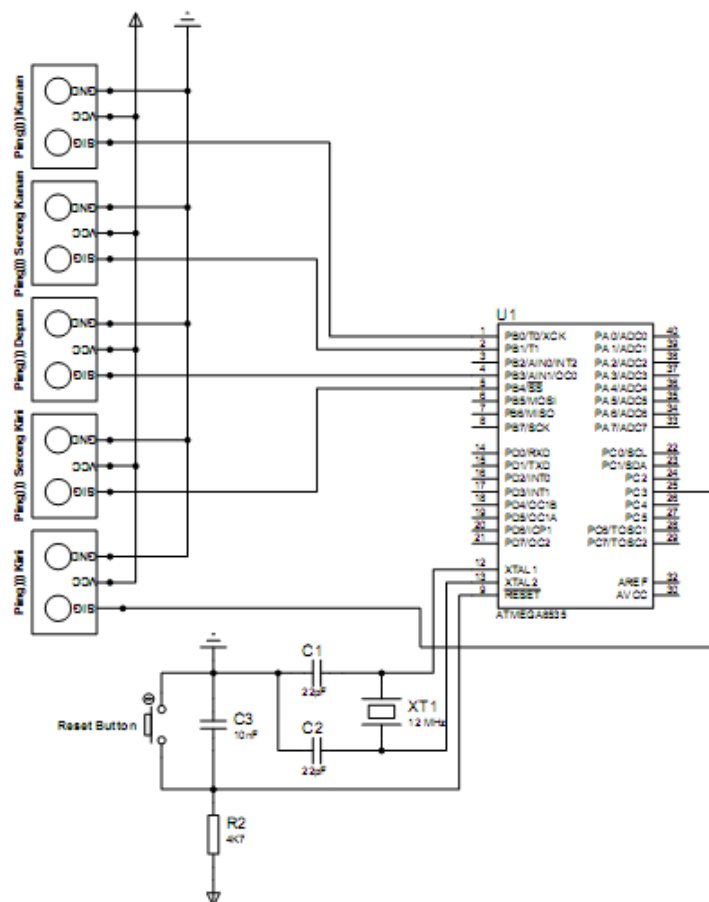
Pada saat check api, jika tidak terpenuhi maka robot berhenti, jika terpenuhi robot akan menyalakan servo dan kipas selama 2 – 3 detik, kemudian melakukan rutin maju 0,3 detik berhenti dan check api kembali.

## BAB IV IMPLEMENTASI DAN ANALISA

Pada bab ini akan dibahas mengenai langkah-langkah pengujian terhadap sensor Ping Parallax ultrasonic Range Finder. Pengujian dilakukan untuk mengetahui apakah sensor Ping Parallax ultrasonic Range Finder bekerja dengan baik sebagai pengukur jarak pada robot pemadam api. Pengujian sensor Ping Parallax ultrasonic Range Finder dikhususkan pada:

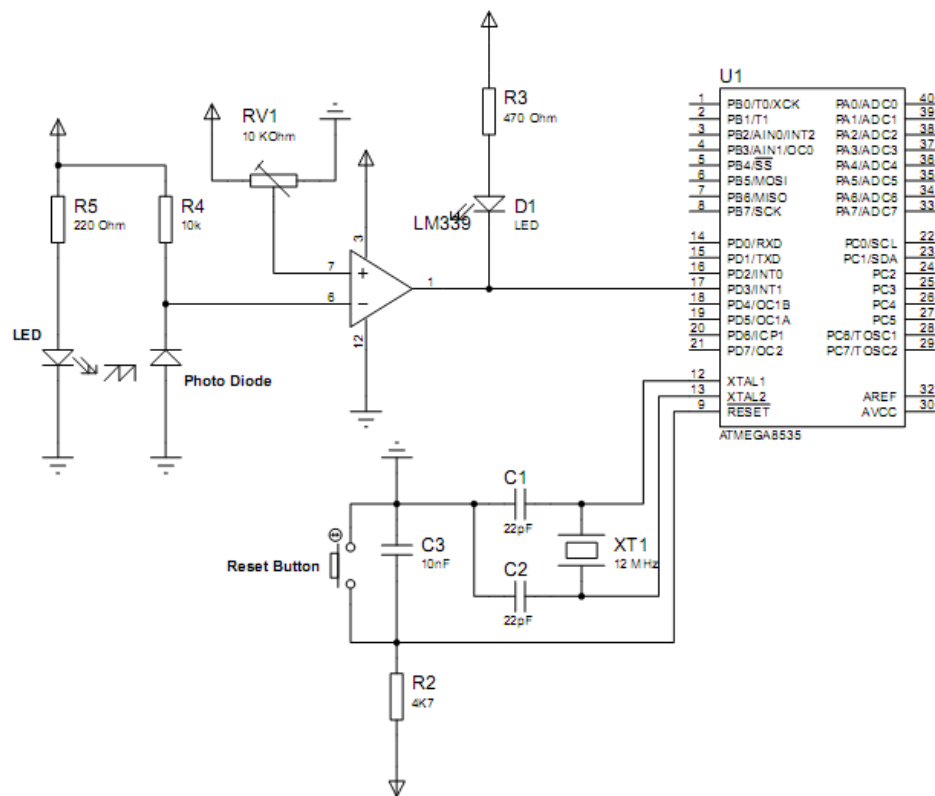
1. Pengujian jarak oleh sensor Ping Parallax Ultrasonic Range Finder.
2. Pengujian jarak pembacaan sensor garis.

Pengujian sensor Ping Parallax ini dengan menghubungkan kaki *sig* ke mikrokontroler selanjutnya menggunakan bantuan LCD untuk menampilkan hasil pembacaan sensor. Titik pengujian ditunjukkan pada Gambar 4.1.



**Gambar 4.1** Rangkaian skematik sensor ultrasonik Ping

Pengujian sensor garis ini akan mengukur jarak pembacaan saat berada pada garis putih. Rangkaian skematik sensor garis ini ditunjukkan pada Gambar



**Gambar 4.2** Rangkaian sensor garis

#### 4.1 Pengujian Jarak Oleh Sensor Ping Parallax Ultrasonic Range Finder

Dari hasil pengujian seperti Table 4.1. terlihat jarak hasil pengujian pada alat tidak sama dengan jarak hasil perhitungan menggunakan rumus. Perbedaan jarak hasil pengujian dengan jarak hasil perhitungan dapat disebabkan oleh adanya noise. Modul sensor PING bekerja berdasarkan prinsip pemantulan gelombang ultrasonik, terkadang pantulan gelombang ultrasonik menjadi tidak periodik dan menyebabkan hasil pengukuran tidak akurat. Selain itu, kesalahan pengukuran juga dapat terjadi karena pembulatan perhitungan pada saat pembuatan program.

**Tabel 4.1.***Hasil Pengukuran Jarak Rata-rata dan Persen Kesalahan*

<b>Jarak Yang Diinginkan</b>	<b>Jarak Hasil Pengukuran Rata-rata</b>	<b>Kesalahan (%)</b>
3 cm	3.02 cm	0.67
4 cm	4.03 cm	0.75
5 cm	5.06 cm	1.20
6 cm	6.11 cm	1.83
7 cm	7.09 cm	1.29
8 cm	8.13 cm	1.62
9 cm	9.07 cm	0.78
10 cm	10.09 cm	0.90
11 cm	11.15 cm	1.36
12 cm	12.26 cm	2.16
13 cm	13.25 cm	2.15
14 cm	14.22 cm	1.57
15 cm	15.27 cm	1.80
16 cm	16.33 cm	2.06
17 cm	17.40 cm	2.35
18 cm	18.37 cm	2.05
19 cm	19.42 cm	2.21
20 cm	20.47 cm	2.35

#### 4.2 Pengujian Jarak Pembacaan Sensor Garis

Robot pemadam ini menggunakan sensor garis untuk mengetahui jarak maksimal antara robot dengan sumber api saat pemadamam api akan dilakukan. Pemasangan sensor pada robot mempunyai ketinggian 2 cm terhadap lantai. Sensor garis yang digunakan pada robot ini merupakan perpaduan antara LED dengan *photodiode*, dimana LED sebagai pemancar dan *photodiode* sebagai penerima.

**Tabel 4.2 Hasil Pengukuran Jarak Sensor Garis**

<b>Pengujian</b>	<b>Jarak</b>	<b>Indikator LED pada Saat di Garis Putih</b>	<b>Indikator LED pada Saat Tidak di Garis Putih</b>
1	1 cm	Hidup	Mati
2	1.5 cm	Hidup	Mati
3	2 cm	Hidup	Mati
4	2.5 cm	Hidup	Mati
4	3 cm	Hidup	Mati
5	3.5 cm	Hidup	Mati
6	4 cm	Hidup	Mati
8	4.5 cm	Hidup	Mati
9	5 cm	Mati	Mati
10	5.5 cm	Mati	Mati

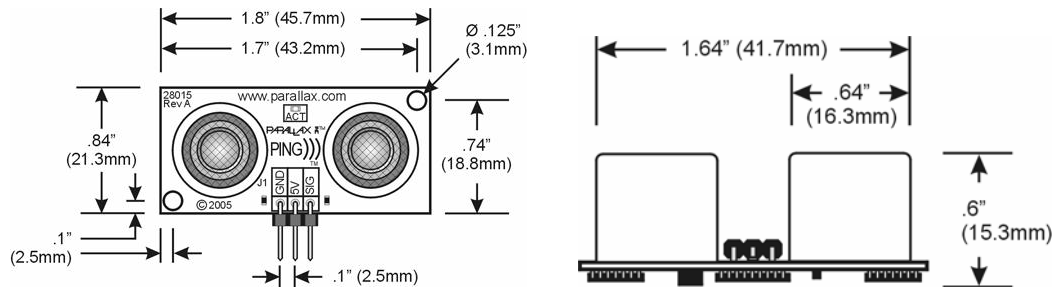
### 4.3 Analisa Pengujian Sensor Ping Parallax dan Sensor Garis

#### 4.3.1 Analisa Pengujian Sensor Ping Parallax

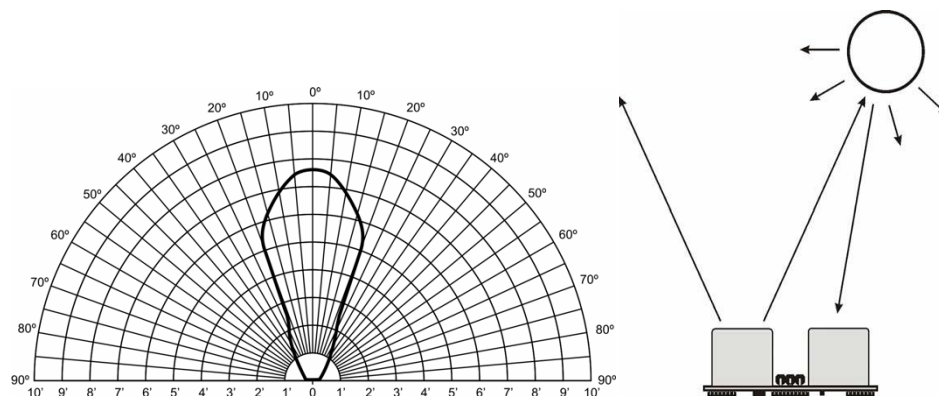
Sensor Ping ini akan mendeteksi jarak suatu objek yang berada didepannya dengan cara memancarkan gelombang ultrasonik dan setelah gelombang itu dipantukan oleh objek tersebut maka gelombang ultrasonik ini akan diterima oleh unit sensor penerima. Ping hanya akan mengirimkan suara ultrasonik ketika ada pulsa trigger dari mikrokontroler. Selanjutnya pulsa ini akan dikirimkan sensor ping ke mikrokontroler melalui *port sig* dan akan diproses oleh mikrokontroler untuk ditampilkan melalui LCD. Ping ini tidak dapat mengukur objek yang permukaannya dapat menyerap suara, seperti busa. Pengukuran jarak juga akan kacau jika permukaan objek bergerigi dengan sudut tajam.

Tegangan keluaran dari sensor digunakan sebagai tegangan input pada komparator. Adapun tegangan yang digunakan sebagai referensi untuk

menentukan tegangan output berupa 0 dan 1 adalah tegangan 5 volt. Tegangan tersebut merupakan tegangan standart dari sensor ultrasonic tersebut.

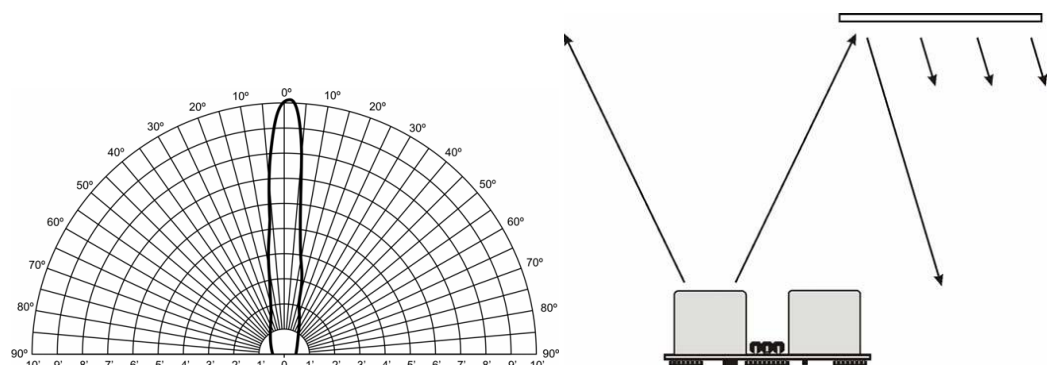


**Gambar 4.3 Dimensi Sensor Ultrasonik**



**Gambar 4.4 Pengujian Ping dengan Halangan Tabung**

Pada pengujian sensor ping pada Gambar 4.4 dengan menggunakan penghalang berupa silinder atau tabung maka jarak maksimal yang diukur akan lebih pendek daripada menggunakan penghalang dari papan. Pengujian diatas terlihat bahwa pada jarak 7,5 inchi sudut pembacaan sensor ping ini hanya mampu pada sudut 0°. Sedangkan sudut pembacaan maksimal yaitu 45 ° pada jarak 1 inchi.



**Gambar 4.5 Pengujian Ping dengan Halangan Papan**

(Sumber: Datasheet PING)))<sup>TM</sup> Ultrasonic Distance Sensor, 2006)

Hasil pengujian pada Gambar 4.5 sangat berbeda dengan pengujian pada Gambar 4.4. Pada pengujian ini terlihat bahwa jarak yang dapat dijangkau oleh sensor ping akan lebih panjang dibandingkan dengan menggunakan penghalang tabung, tetapi sudut pembacaan sensor ping akan lebih kecil dibandingkan menggunakan tabung.





**Gambar 4.6** *Flowchart mencari room*

Pengujian ini bertujuan untuk mengetahui hasil pembacaan sensor Ping parallax dapat diproses menggunakan Mikrokontroler ATmega8535 untuk dijadikan *output*. Pengujian dilakukan dengan cara menghubungkan LCD ke port A ATmega8535, pin data sensor ping parallax dihubungkan ke port B.0 ATmega8535. Mikrokontroler ATmega8535 diisi program berikut:

```
#include <mega8535.h>
#include <delay.h>
#include <stdio.h>
// Alphanumeric LCD Module functions
#asm
.equ __lcd_port=0x1B ;PORTA
#endasm
#include <lcd.h>
// Declare your global variables here
#define SigOut PORTB.0
#define SigIn PINB.0
#define DirSig DDRB.0
unsigned int counter;
floate;
distancunsigned char Baris1[16];
unsigned char Baris2[16];
void main(void)
{
// Declare your local variables here
// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
```

```
PORTA=0x00;
DDRA=0x00;
// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTB=0x00;
DDRB=0x00;
// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTC=0x00;
DDRC=0x00;
// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTD=0x00;
DDRD=0x00;
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
```

```
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
```

```

MCUCR=0x00;
MCUCSR=0x00;
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;
// LCD module initialization
lcd_init(16);
while (1)
{
// Place your code here
counter=0; //Initial value
DirSig=1; //Set as output
SigOut=1;
delay_us(5);
SigOut=0;
DirSig=0; //Set as input
SigOu while (SigIn==0)
t=1; //Pullup activated
{ }
while (SigIn==1)
{counter++;}
// V=344.424m/s
distance=(counter*0.034442)/1.21; // Untuk clock 12 Mhz
sprintf(Baris1,"Counter=%5d uS",counter);
sprintf(Baris2,"Jarak: %3f cm",distance);
lcd_gotoxy(0,0);lcd_puts(Baris1);
lcd_gotoxy(0,1);lcd_puts(Baris2);

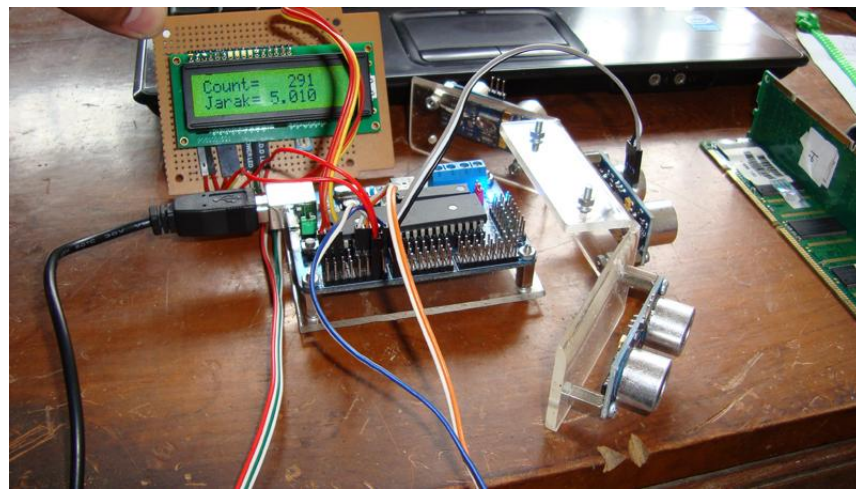
```

```

delay_ms(200);
lcd_clear();
};
}

```

Program diatas ketika dijalankan pada LCD akan menampilkan pembacaan sensor Ping, pembacaan ini berupa *count* dan jarak yang terbaca oleh sensor ping. Hasil pengujian sensor ini dapat dilihat pada Gambar 4.7.



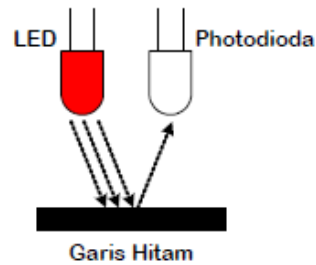
**Gambar 4.7** Pengujian Ping Ultrasonic

#### 4.3.2 Analisa Pengujian Sensor Garis

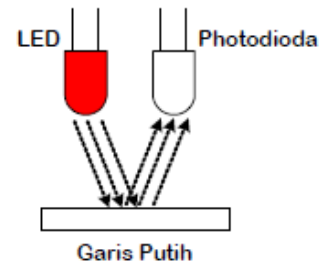
Pengujian sensor garis yang menggunakan IC LM339 ini dapat mendeteksi adanya garis putih sampai dengan jarak 4.5 cm. Keluaran dari sensor ini bernilai *high* (1) dengan tegangan *output* 5 v pada saat sensor berada pada garis putih, sebaliknya sensor akan bernilai *low* (0) dengan tegangan *output* 0 v pada saat sensor tidak berada pada garis putih.

Prinsip kerjanya sederhana, hanya memanfaatkan sifat cahaya yang akan dipantulkan jika mengenai benda berwarna terang dan akan diserap jika mengenai benda berwarna gelap. Sebagai sumber cahaya kita gunakan LED (*Light Emitting Diode*) yang akan memancarkan cahaya merah. Dan untuk menangkap pantulan cahaya LED, kita gunakan photodiode. Jika sensor berada diatas garis hitam maka photodiode akan menerima sedikit sekali cahaya pantulan. Tetapi jika sensor

berada diatas garis putih maka photodioda akan menerima banyak cahaya pantulan. Berikut adalah ilustrasinya:



Gambar 1. Cahaya pantulan sedikit



Gambar 2. Cahaya pantulan banyak

#### **Gambar 4.8** *Prinsip Kerja Sensor Garis*

Sifat dari photodioda adalah jika semakin banyak cahaya yang diterima, maka nilai resistansi diodanya semakin kecil. Dengan melakukan sedikit modifikasi, maka besaran resistansi tersebut dapat diubah menjadi tegangan. Sehingga jika sensor berada diatas garis hitam, maka tegangan keluaran sensor akan kecil, demikian pula sebaliknya.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Kesimpulan dari pembuatan robot pemadam api adalah sebagai berikut :

1. Telah dibuat robot pemadam api dengan panjang 24 cm, lebar 15 cm dan tinggi 17 cm. Robot pemadam api menggunakan sensor Ping Ultrasonik, sensor kompas dan sensor garis sebagai sistem navigasi, sensor UVTron sebagai pendeteksi api dan sistem gerak menggunakan motor DC.
2. Sensor Ping Ultrasonik digunakan untuk navigasi pada robot. Sensor ini akan mengukur jarak robot terhadap dinding pada ruangan, hasil pembacaan sensor Ping Ultrasonik ini melalui port *sig* yang dihubungkan dengan *port* pada mikrokontroler selanjutnya ditampilkan melalui LCD.
3. Sensor Garis yang digunakan dapat mendeteksi adanya garis putih. Pembacaan sensor ini berupa nilai *high* (1) pada saat berada digaris putih, sedangkan bernilai *low* (0) saat belum berada pada garis putih.
4. Robot pemadam ini menggunakan metode *left wall following* untuk mencari ruangan pertama dengan bantuan sensor Ping Ultrasonik kemudian menggunakan bantuan sensor kompas untuk berbelok atau berputar, dan menggunakan sensor UVTron untuk mendeteksi sumber api pada ruangan tersebut.

#### **5.2 Saran**

Beberapa saran yang perlu disampaikan pada pembuatan robot pemadam api ini adalah:

1. Tingkat keakuratan sensor Ping Parallax Ultrasonic Range Finder dapat dilakukan dengan melalui pemrograman mikrokontroler. Selain itu untuk mendapatkan hasil yang lebih akurat pada rangkaian minimum sistem menggunakan kristal 11,0592 MHz. Sedangkan pemakaian kristal pada minimum sistem umumnya menggunakan 12 MHz. Selain itu objek yang

digunakan untuk memantulkan harus berupa benda padat agar sinyal yang dipancarkan dapat diterima kembali secara baik.

2. Penyaluran catu daya pada masing-masing komponen dan sensor belum merata, sehingga terjadi *drop* (penurunan) daya pada saat semua komponen dan sensor bekerja secara bersamaan.
3. Dalam pemograman robot pemadam ini belum terdapat algoritma untuk mengingat ruangan mana yang telah diperiksa dan ruangan mana yang belum diperiksa.



## DAFTAR PUSTAKA

Averroes, Fitra Luthfie. 2009. *Tugas Akhir: Rancang Bangun Robot Pemadam Api Berbasis Mikrokontroler ATmega8535*. Diploma III Ilmu Komputer Universitas Sebelas Maret: Surakarta

Budiharto, Widodo. 2008. *10 Proyek Robot Spektakuler*. Jakarta: Elex Media Komputindo

Heryanto, Ary dan Wisnu, Adi. 2008. *Pemrograman Bahasa C untuk Mikrokontroler ATMEGA8535*. Yogyakarta: Andi Offset

Kurniawan, Degi. 2009. *Tugas Akhir: Aplikasi Mikrokontroler ATmega8535 Pada Robot Pemadam Api*. Diploma III Teknik Elektro Politeknik Negeri Sriwijaya: Palembang

Pitowarno, Endra. 2006. *Robotika Desain, Kontrol, dan Kecerdasan Buatan*. Yogyakarta: Andi Offset

Purwantoni, Budi. 2009. *Proyek Akhir: Rancang Bangun Saklar Otomatis Berbasis Waktu Untuk Mengontrol Perangkat Elektronik Rumah Tangga Dengan Mikrokontroler Atmega 8535*. Teknik Elektro Institut Teknologi Telkom: Bandung

Wardhana, Lingga. 2006. *Belajar Sendiri Mikrokontroler AVR Seri ATmega8535 Simulasi, Hardware, dan Aplikasi*. Yogyakarta: Andi Offset

# **LAMPIRAN 1**

## LISTING PROGRAM SECARA KESELURUHAN

/\*\*\*\*\*\*

This program was produced by the  
CodeWizardAVR V2.03.4 Standard  
Automatic Program Generator  
© Copyright 1998-2008 Pavel Haiduc, HP InfoTech s.r.l.  
<http://www.hpinfotech.com>

Project : Robot 1  
Version : Beta 1  
Date : 7/13/2010  
Author : Harnan Amri  
Company : DIII Teknik Komputer UNS  
Comments:

Chip type : ATmega8535  
Program type : Application  
Clock frequency : 12.000000 MHz  
Memory model : Small  
External RAM size : 0  
Data Stack size : 128

\*\*\*\*\*/

```
#include <mega8535.h>
#include <mega8535.h>
#include <delay.h>
#include <stdio.h>
```

```
//=====DEFINE PORT=====//
```

```
#define ping1out PORTB.0
#define ping1in PINB.0
#define dir1ping DDRB.0
#define ping2out PORTB.1
#define ping2in PINB.1
#define dir2ping DDRB.1
#define enka PORTD.4
#define enki PORTD.5
#define in1ka PORTD.0
#define in2ka PORTD.1
#define in3ki PORTD.6
#define in4ki PORTD.7
```

```
//=====VARIABEL=====//
```

```
unsigned int count1, count2;
float jarak1, jarak2;
unsigned char baris1[16];
unsigned char baris2[16];
unsigned char flame;
unsigned char garis;
unsigned char n1,n2;
int i;
```

```

// Alphanumeric LCD Module functions
#asm
.equ __lcd_port=0x1B ;PORTA
#endasm
#include <lcd.h>
//=====INTERUPSI GARIS=====//
// External Interrupt 1 service routine
interrupt [EXT_INT1] void ext_int1_isr(void)
{
// Place your code here
garis = 1;
}
//=====INTERUPSI API=====//
// External Interrupt 2 service routine
interrupt [EXT_INT2] void ext_int2_isr(void)
{
// Place your code here
flame = 1;
}
// Declare your global variables here
//=====BACA PING=====//
void ping1(){
count1=0;
    dir1pin=1;
    ping1out=1;
    delay_us(5);
    ping1out=0;
    dir1pin=0;
    ping1out=1;

    while(ping1in==0){};
    while(ping1in==1){
        count1++;
    };
    jarak1=(count1*0.034442)/2;
    sprintf(baris1,"%3.0f",jarak1);
    lcd_gotoxy(0,0); lcd_puts(baris1);
    delay_ms(20);
}

void ping2(){
count2=0;
    dir2pin=1;
    ping2out=1;
    delay_us(5);
    ping2out=0;
    dir2pin=0;
    ping2out=1;

    while(ping2in==0){};
    while(ping2in==1){
        count2++;
    };
    jarak2=(count2*0.034442)/2;

```

```

    sprintf(baris2,"%3.0f",jarak2);
    lcd_gotoxy(0,1); lcd_puts(baris2);
    delay_ms(20);
}
//=====GERAK MOTOR=====//
void maju(){
in1ka=1;in2ka=0;
in3ki=1;in4ki=0;
}
void mundur(){
in1ka=0;in2ka=1;
in3ki=0;in4ki=1;
}
void pu_ka(){
in1ka=0;in2ka=1;
in3ki=1;in4ki=0;
}
void pu_ki(){
in1ka=1;in2ka=0;
in3ki=0;in4ki=1;
}
void stop(){
in1ka=0;in2ka=0;
in3ki=0;in4ki=0;
}
//=====HIDUPKAN KIPAS=====//
void kipas(){
PORTC.3 = 1;
}
//=====HIDUPKAN SERVO=====//
void servo(){
n1 = 3;
n2 = 190;
for(i=0;i<n2;i++){
PORTC.2=1;
delay_us(2000);
PORTC.2=0;
delay_us(3000);
}
for(i=0;i<n2;i++){
PORTC.2=1;
delay_us(550);
PORTC.2=0;
delay_us(3000);
}
}
//=====GABUNG PING=====//
void gabung(){
ping1();
ping2();
}

```

```

//=====PADAM API=====//
void padam(){
  kipas();
  servo();
}
//=====SELESAI=====//
void selesai(){
  PORTC.3=0;
  mundur();
  delay_ms(1000);
  stop();
  lcd_clear();
  lcd_gotoxy(0,1);
  lcd_putsf("selesai");
  delay_ms(500000);
}
//=====API=====//
void api(){
  maju();
  delay_ms(300);
  stop();
  delay_ms(700);
  if (flame==1){
    padam();
    delay_ms(500);
    flame=0;
    lcd_gotoxy(0,1);
    lcd_putsf("api api api");
  }
  else{
    selesai();
  }
}

//=====BACA SENSOR KIRI=====//
void baca_kiri(){
  gabung();
  maju();
  while ((jarak2 >= 11) && jarak1 >= 10){pu_ki(); gabung();}
  while ((jarak2 <= 7) && jarak1 >= 10){pu_ka(); gabung(); }
  while (jarak1 <= 7) {pu_ka(); delay_ms(1000); maju(); delay_ms(200); ping1(); }
}

void check1(){
  if (flame == 1 && garis==0) {
    baca_kiri();
  }
  else
  if ((flame == 1) && garis == 1){
    api();
  }
}

```

```

void main(void)
{
// Declare your local variables here
// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;
// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x00;
// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=Out Func2=Out Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=0 State2=0 State1=T State0=T
PORTC=0x00;
DDRC=0x0C;
// Port D initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=In Func2=In Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=T State2=T State1=0 State0=0
PORTD=0x00;
DDRD=0xF3;
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

```

```

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
// External Interrupt(s) initialization
// INT0: Off
// INT1: On
// INT1 Mode: Rising Edge
// INT2: On
// INT2 Mode: Rising Edge
GICR|=0xA0;
MCUCR=0x0C;
MCUCSR=0x40;
GIFR=0xA0;
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;
// LCD module initialization
lcd_init(16);

// Global enable interrupts
#asm("sei")

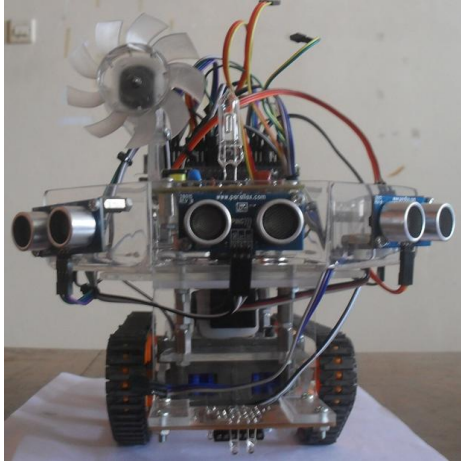
while (1)
{
    // Place your code here
    check1();
};
}

```

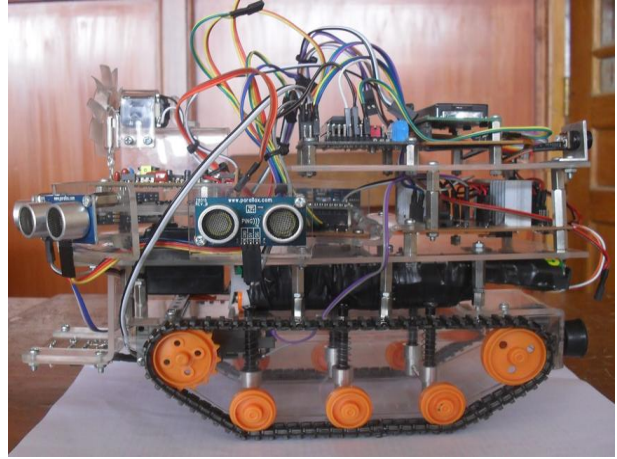


## **LAMPIRAN 2**

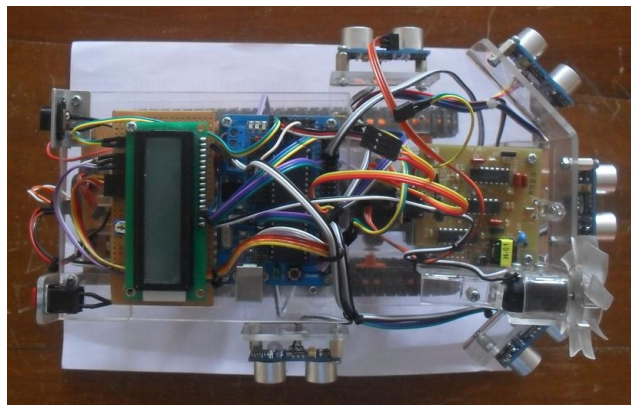
## BENTUK FISIK ROBOT



**Tampak Depan**



**Tampak Samping Kiri**



**Tampak Atas**